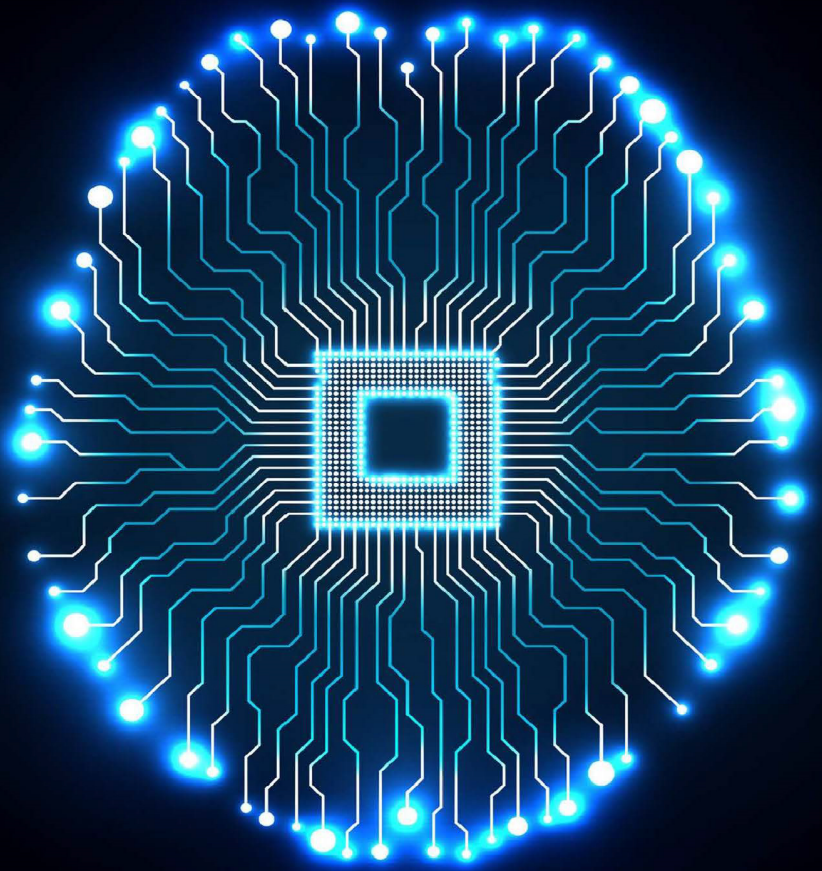
MATRIX | EBLOCKS 2

FLOWCODE

Introduction to microcontroller programming

Suitable for BTEC National in Engineering unit 6,
Microcontroller Systems for Engineers



CP4375-3

MATRIX

www.matrixtsl.com

Contenido

Introducción

El objetivo de este curso es introducirle en los conceptos de desarrollo de sistemas electrónicos utilizando microcontroladores.

De este modo, ofrece una cobertura sustancial de **la Unidad 6 del BTEC Level 3 National Extended Diploma in Engineering** (la correspondencia precisa del curso con esta unidad figura en la página 9).

Al finalizar este curso habrás aprendido:

- qué es un microcontrolador.
- cómo construir circuitos y sistemas basados en microcontroladores.
- cómo programar microcontroladores.

Antes de empezar

Este curso es una introducción a la programación de microcontroladores.

Para aprovechar al máximo este curso, le recomendamos que tenga lo siguiente:

Flowcode

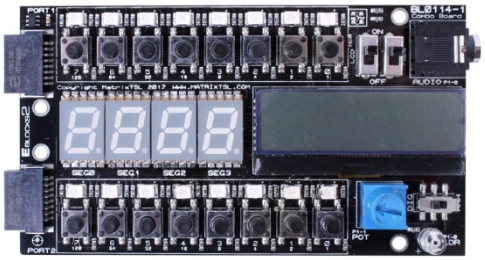
Flowcode es un programa de software que permite a los usuarios desarrollar de forma rápida y sencilla sistemas electrónicos complejos. Funciona con una amplia gama de microcontroladores, incluidos los microcontroladores 'PIC' de Microchip (PIC MCU), Arduino y ARM. Flowcode es neutro con respecto a los microcontroladores: presenta prácticamente la misma interfaz de usuario con independencia del microcontrolador utilizado. Las diferencias radican en el hardware y en la forma de descargar y probar el programa.

Hardware

Siempre es más gratificante cuando se aprende acerca de la programación de microcontroladores ver los programas ejecutarse en hardware real, por lo tanto, le recomendamos que tenga algún hardware disponible para enviar y ejecutar sus programas creados.

Este curso está diseñado principalmente en torno a la plataforma de hardware Matrix E-blocks2, normalmente el programador BL0011 y la placa BL0114 Combo, aunque también se pueden utilizar E-blocks independientes (LCD, interruptores, LED, etc.).

Programador BL0011.



BL0114 Tablero combinado.

Aunque la mayor parte del curso está diseñado en torno a los E-blocks2, también reconocemos que algunas personas pueden estar utilizando ya sea un dispositivo Arduino. Así que en este curso, siempre que haya cambios en las instrucciones para Arduino cambios, se mostrarán en los siguientes colores:

Los usuarios de Arduino necesitan un Arduino Uno y E-blocks Arduino Uno Shield (BL0055), así como la placa Combo.

Más información

Flowcode

https://www.flowcode.co.uk

Desde aquí puede acceder:

- Flowcode-Guía de inicio

- Wiki de Flowcode

- Una amplia gama de ejemplos Flowcode

Bloques E

https://www.matrixtsl.com/eblocks/resources

En la sección E-blocks puede obtener los siguientes recursos:

- Controladores USB E-blocks

- Ficheros de ejemplo de bloques electrónicos

- E-blocks Guía del usuario

https://www.matrixtsl.com/eblocks/boards

Desde las páginas de los tableros:

- Fichas técnicas específicas de las tarjetas

- Ejemplos concretos de juntas directivas

Otra ayuda

https://www.flowcode.co.uk/forums/

El foro Matrix ofrece una comunidad de usuarios de Flowcode consolidados y veteranos, así como de nuevos usuarios de Flowcode, que comparten ideas y resuelven problemas y cuestiones que surgen al utilizar el software.

https://www.matrixtsl.com/learning/

El "Centro de Aprendizaje" de Matrix contiene muchos recursos diferentes, como artículos, controladores y planes de estudios.

Convenciones del curso

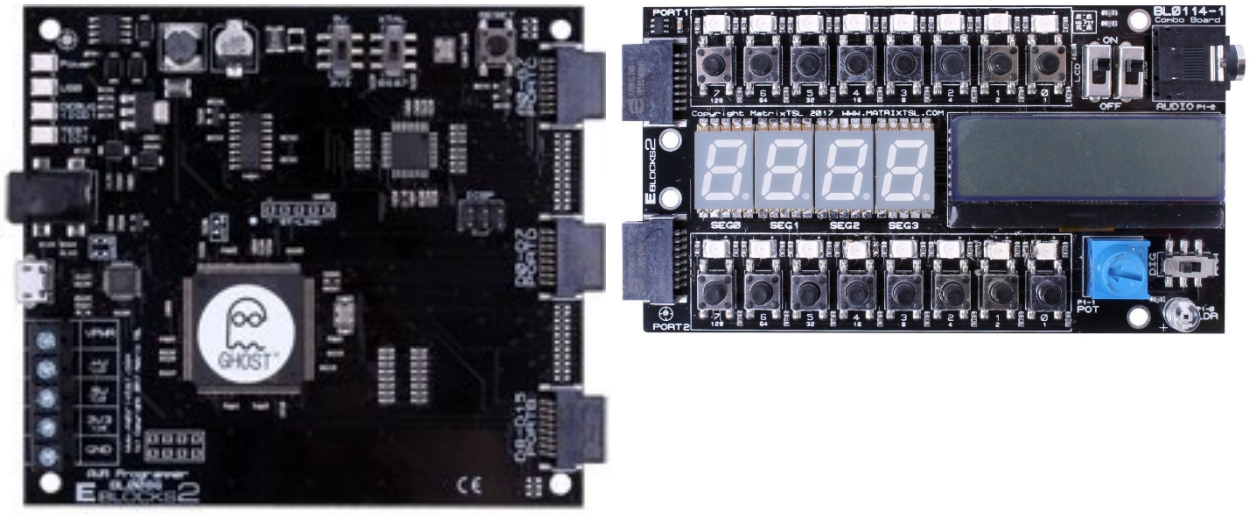
En el curso se utilizan las siguientes abreviaturas:

Abbreviation	Meaning
ADC	**A**nalogue to **D**igital **C**onverter
ALU	**A**rithmetic **L**ogic **U**nit
ASCII	**A**merican **S**tandard **C**ode for **I**nformation **I**nterchange
CPU	**C**entral **P**rocessing **U**nit
EEPROM	**E**lectrically **E**rasable **P**rogrammable **R**ead **O**nly **M**emory
EPROM	**E**rasable **P**rogrammable **R**ead **O**nly **M**emory
GND	ground
Hex	hexadecimal
IDC	**I**nsulation **D**isplacement **C**onnector
I/O	**I**nput / **O**utput
ISP	**I**n-**S**ystem **P**rogramming
JPEG	**J**oint **P**icture **E**xpert **G**roup (standard for images)
LCD	**L**iquid **C**rystal **D**isplay
LED	**L**ight **E**mitting **D**iode
LVP	**L**ow **V**oltage **P**rogramming
LDR	**L**ight **D**ependent **R**esistor
LSB	**L**east **S**ignificant **B**it
MSB	**M**ost **S**ignificant **B**it
NVRAM	**N**on-**V**olatile **R**andom **A**ccess **M**emory
PIC	**P**eripheral **I**nterface **C**ontroller
PROM	**P**rogrammable **R**ead **O**nly **M**emory
PSU	**P**ower **S**upply **U**nit
RAM	**R**andom **A**ccess **M**emory
RV1	**R**esistor-**V**ariable **1**
SPI	**S**erial **P**rogrammable **I**nterface
XTAL	crystal
ZIF	**Z**ero **I**nsertion **F**orce
+V	positive supply voltage

Introducción

El hardware:
En la mayoría de los ejercicios se utiliza el multiprogramador BL0011 / BL0080 y la placa combinada BL0114.

La mayoría de los ejercicios también pueden realizarse utilizando el Arduino Uno Shield (BL0055). Sin embargo, estos requieren diferentes configuraciones de PUERTO.



Ajustes de hardware y software utilizados para probar la mayoría de los programas:

BL0011 / BL0080		Switch Settings
Microchip MCU	16F18877	
Target Voltage	5V	5V
Oscillator Selection	External Oscillator	XTAL
Port A E-Block	E-Blocks Combo board BL0114	
Port B E-Block		
Port C E-Block		
Port D E-Block		
Port E E-Block		

Flowcode y ajustes de descarga:

Menu	Name	Settings	Eblocks 1
Build > Project Options... > Choose a Target	8-bit PIC	16F18877	16F1937
Build > Project Options... > Configure	External Oscillator Mode	HS Oscillator, High speed crystal	N/A
	Software Oscillator Mode	EXTOSC with 4X PLL	N/A
	Oscillator	N/A	HS Oscillator
	WDT Operation Mode	WDT disabled, SWDTEN is ignored	N/A
	WDT Input Selector	Software Control	N/A
	Watchdog Timer Enabled Bit	N/A	Disabled
Other Options	Clock speed (MHz)	32	19.6608
	Simulation Speed	Fast (No Updates)	Fast (No Updates)

Nota: Cuando se seleccionan objetivos BL0011, BL0080 o 16F18877 en Flowcode, los ajustes por defecto son para el Oscilador interno .

Después de completar este curso, serás capaz de:

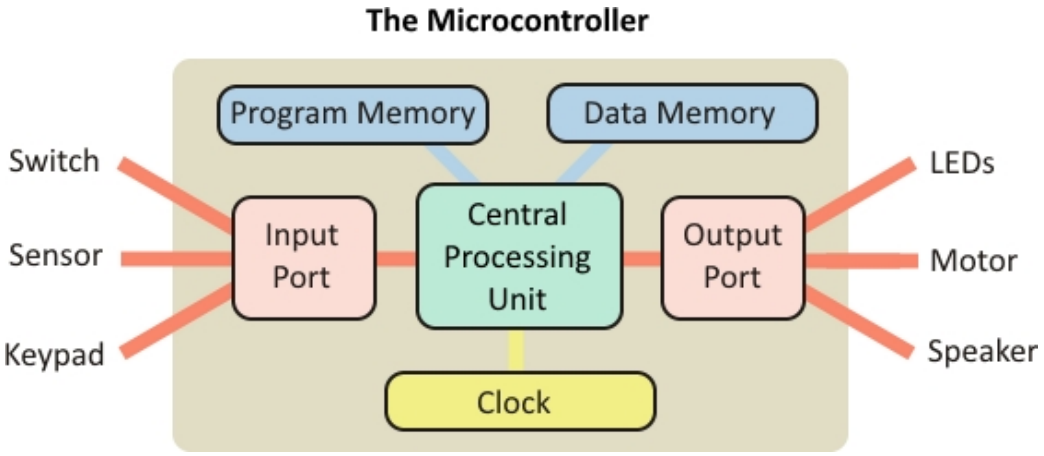
1. Envía diferentes códigos de 8 bits a los puertos del microcontrolador.
2. Cambia el nivel lógico de un solo pin.
3. Configura un icono de salida.
4. Utiliza código binario.
5. Manipula los niveles lógicos de salida.
6. Utiliza los LED para visualizar una salida.
7. Compilar un programa en el PIC MCU.
8. Añade un retardo para ralentizar la ejecución de un programa.
9. Cambia el intervalo de retardo.
10. Configura un icono de retardo.
11. Controla la velocidad de un microcontrolador.
12. Utiliza un osciloscopio para cronometrar los eventos.
13. Utilice los puntos de conexión para introducir bifurcaciones incondicionales en un programa.
14. Introducir PWM como medio para controlar el brillo de los LED.
15. Crea un bucle infinito.
16. Manipula los niveles lógicos de salida.
17. Utiliza LEDs para visualizar una salida.
18. Crear y utilizar una variable.
19. Configura un icono de cálculo para realizar cálculos aritméticos y lógicos.
20. Crear y manipular variables.
21. Realiza cálculos.
22. Utiliza LED con resistencias limitadoras de corriente.
23. Crea y utiliza un programa de "semáforo en marcha" utilizando el método de "multiplicar por dos".
24. Crea y utiliza un programa de "luz de marcha" utilizando el método "shift-right".
25. Crear y rellenar un array.
26. Crear un bucle condicional.
27. Datos de entrada de los interruptores.
28. Utiliza bucles para crear secuencias de LED.
29. Configurar un icono de entrada.
30. Configurar iconos de decisión y, por tanto, añadir bifurcaciones condicionales a un programa.
31. Controla la frecuencia con la que parpadean los LED.
32. Utiliza LEDs para mostrar los niveles lógicos de salida.
33. Utilizar memoria temporal.
34. Crear, rellenar y manipular variables de cadena.
35. Controla la visualización de texto y números en una pantalla LCD.
36. Utilizar una pantalla LCD como dispositivo de salida para el PIC MCU.
37. Configure una macro Componente para la pantalla LCD.
38. Introduzca texto y números desde un teclado y visualice mensajes en la pantalla LCD.
39. Utilice el código ASCII para transmitir estos datos.
40. Utiliza entradas multiplexadas.
41. Configurar una macro Componente para el teclado.
42. Crear registradores de datos, utilizando datos de 8 y 10 bits del ADC.
43. Configura una entrada analógica.
44. Introducir datos mediante interruptores.
45. Introduce la información de los sensores de luz y temperatura.
46. Configurar y utilizar la EEPROM.
47. Desplazarse por los datos de la EEPROM.
48. mostrar texto y datos numéricos en la pantalla LCD.
49. Utiliza el tablero de prototipos E-blocks.
50. Utilice macros de software para simplificar la estructura de un programa.
51. Crear macros de software.
52. Utilice el control de bucle cerrado.
53. Utiliza PWM para controlar el brillo de los LEDs.
54. Crea y utiliza interrupciones de un solo pin.
55. Crear y utilizar interrupciones por cambio (IOC).
56. Utiliza el funcionamiento en tiempo real de una MCU PIC.
57. Crear y utilizar interrupciones de temporizador.
58. Utiliza el preescalador para crear intervalos de tiempo precisos.
59. Dispara el temporizador utilizando el cristal o un evento externo.

Sección 1:

Introducción a la Microcontroladores

Los microcontroladores son dispositivos diminutos que se utilizan para controlar otros dispositivos electrónicos. Se encuentran en una amplia gama de productos. En sistemas de automoción están presentes en motores, frenos antibloqueo y sistemas de climatización. En la electrónica doméstica están presentes en televisores, videograbadoras, cámaras digitales, teléfonos móviles, impresoras, hornos microondas, lavavajillas y lavadoras.

Un **microcontrolador** es un circuito integrado digital que consta de una unidad central de procesamiento, una memoria, puertos de entrada y puertos de salida.



En su corazón (¿o es su cerebro?) hay una Unidad Central de Procesamiento (CPU). Esta procesa las señales digitales, realiza cálculos y operaciones lógicas, crea retardos, establece secuencias de señales, etc.

¿Cómo sabe lo que tiene que hacer? Sigue un programa de instrucciones, almacenado en una parte de la memoria, llamada "memoria de programa", dentro del PIC.

De vez en cuando, la CPU necesita almacenar datos para recuperarlos más tarde. Para ello, utiliza otra zona de la memoria, denominada "memoria de datos".

El reloj sincroniza las actividades de la CPU. Envía un flujo de impulsos de tensión a la CPU que controla cuándo se mueven los datos por el sistema y cuándo se ejecutan las instrucciones del programa. Cuanto más rápido funcione el reloj, más rápido ejecutará el microcontrolador el programa. Normalmente, el reloj funciona a una frecuencia de 20 MHz (veinte millones de impulsos de tensión por segundo).

Para comunicarse con el mundo exterior, el microcontrolador dispone de "puertos" que introducen o envían datos en forma de números binarios. Cada puerto tiene un número de conexiones, a menudo denominadas "bits". Un puerto de 8 bits gestiona un número de 8 bits (o un byte).

La información procedente de los sensores se introduce en el sistema a través de los puertos de entrada. El microcontrolador procesa estos datos y los utiliza para controlar los dispositivos conectados a los puertos de salida. Los puertos son circuitos electrónicos complejos, no simples terminales para colgar componentes.

¿Qué es un microcontrolador?

Microcontroladores - PIC y AVR

El nombre PIC, (Peripheral Interface Controller), se refiere a un grupo de microcontroladores, producidos por Arizona Microchip.

Cuando utilizamos un microcontrolador PIC, tenemos que especificar cómo queremos que se comporten los puertos. Los puertos son bidireccionales, lo que significa que pueden actuar como puertos de entrada o como puertos de salida. Cuando escribimos un programa para el PIC, empezamos configurando los puertos, diciéndoles si deben comportarse como puertos de entrada o de salida.

El puerto de entrada puede recibir datos (información) en una de dos formas, como analógico o como señal digital. Es importante que entendamos claramente la diferencia entre ambas.

El mundo digital

Gran parte de nuestra información cotidiana se describe en formato numérico. Por ejemplo:

- "Son las 2 en punto."
- "La temperatura exterior es de 21 grados C".
- "El coche circulaba a 48 kilómetros por hora". Es fácil entender los datos de esta forma.

Por ejemplo, la tabla muestra cómo cambia la velocidad de un coche a lo largo de un periodo de tiempo.

Tiempo en segundos	Velocidad en kilómetros por hora
0	0
10	15
20	21
30	25
40	22
50	20
60	16

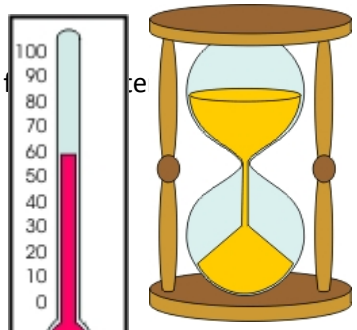
Sin embargo, cabe preguntarse qué ocurrió en el tiempo 35 segundos. ¿Se movía el coche más rápido o más despacio que a 25 km/h en ese momento?

El mundo analógico

Ahora la información se da en forma de analogía. Es decir, utilizamos algo que se comporta de forma similar.

Por ejemplo:

1. El reloj de arena:



Cuanto mayor sea el tiempo transcurrido, mayor será la profundidad de la arena en el f ce

2. El termómetro de mercurio en vidrio

Cuanto más se calienta, más sube el mercurio por el tubo.

3. El velocímetro del coche

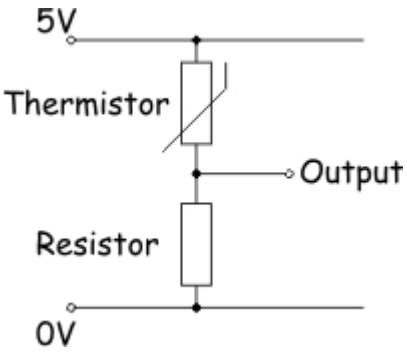
Cuanto mayor sea la velocidad, más se desplazará la aguja por el dial.

El problema de los datos analógicos es que hay que trabajar un poco para extraerlos.

Para el velocímetro y el termómetro, tienes que averiguar dónde se sitúa la aguja. en la escala. Por otro lado, es fácil juzgar cómo está cambiando la temperatura de un cuerpo o la velocidad de un coche: observa lo rápido que se mueve el mercurio a lo largo del tubo o lo rápido que se desplaza la aguja alrededor de la esfera.

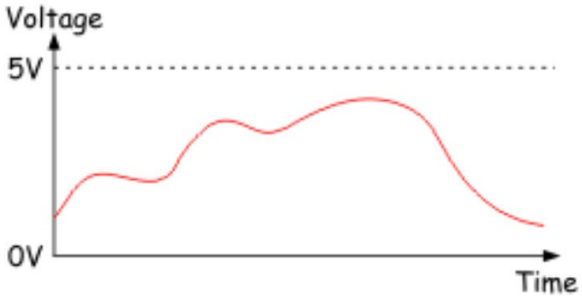
Datos analógicos

Muchos sensores electrónicos proporcionan señales en forma analógica. Por ejemplo, un micrófono proporciona una "copia" eléctrica de una onda sonora.



Otro: ¡el sensor de temperatura!

Este es el esquema de un tipo de sensor de temperatura. La tensión de salida aumenta cuando aumenta la temperatura. Se trata de una señal analógica porque la tensión copia el comportamiento de la temperatura.



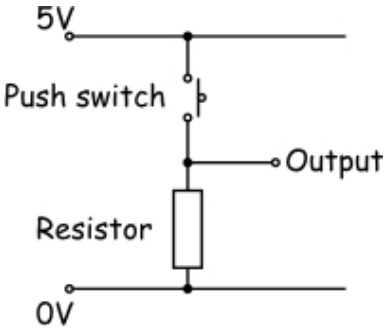
Una señal eléctrica analógica puede tener cualquier valor de tensión, limitado únicamente por la fuente de alimentación utilizada.

En este caso, la salida del sensor de temperatura podría, en teoría, llegar hasta 5V, o hasta 0V.

Durante un periodo de tiempo, la tensión de salida podría cambiar como se muestra en el diagrama. Se trata de una señal analógica.

Datos digitales

Una señal digital transporta su información en forma de número. Los sistemas electrónicos suelen emplear el sistema numérico binario, que sólo utiliza los números "0" y "1", codificados como tensiones. Podríamos optar por el siguiente código '0'= 0V, '1'= 5V, por ejemplo.



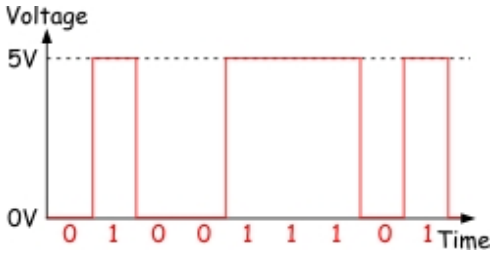
Las señales digitales, por tanto, sólo tienen dos valores de tensión posibles, normalmente la tensión de alimentación, o lo más cercano a ella que pueda conseguir el sistema, y 0V.

¿Cómo podemos introducir estas cifras en un sistema electrónico?

Una forma (muy lenta) sería utilizar un interruptor (un ejemplo de sensor digital.) El diagrama del circuito muestra un sensor digital de este tipo.

- Cuando el interruptor está abierto (no pulsado), la salida es "bajada" a 0V por la resistencia. Esta salida podría representar el número binario '0'.
- Con el interruptor cerrado (pulsado), la salida se conecta a la alimentación positiva, 5V en este caso. Esto podría representar el número binario '1'.

(Nota - si se invirtieran las posiciones del interruptor y la resistencia, al pulsar el interruptor se pondría una señal lógica 0 en el pin, etc.) El siguiente diagrama muestra una señal digital más compleja.



El número binario de nueve bits representado por la señal se indica bajo la forma de onda.

Conversión analógica a digital

Gran parte de los datos del "mundo real" son analógicos, pero los ordenadores (incluidos los microcontroladores) sólo pueden procesar datos digitales. Afortunadamente, los microcontroladores suelen contener un subsistema capaz de convertir la información de formato analógico a formato digital. Es lo que se conoce como conversor analógico-digital, que suele abreviarse como "ADC" o "A/D".

El ADC del microcontrolador divide el rango de posibles tensiones analógicas en pasos iguales. Al paso más bajo se le asigna el número "0", y al paso más alto se le asigna el número más alto que el convertidor A/D puede manejar.

Este número más alto viene determinado por la resolución del ADC, que, a su vez, depende del número de "bits" que pueda manejar el circuito interno del ADC. La resolución de los ADC de los PIC es de 8, 10 ó 12 bits.

Por ejemplo, si la mayor tensión analógica es de 5V, y el PIC tiene un ADC de 8 bits:
el número de 8 bits más alto es 1111 1111 (= 255 en

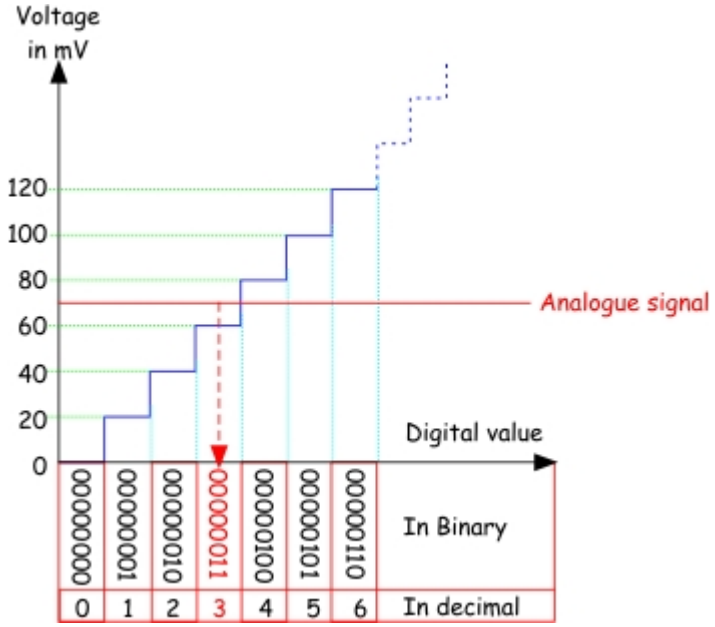
decimal); el primer paso es 0000 0000 (= 0 en decimal) ;

lo que significa que hay 256 niveles de tensión;

por lo que pasar de un nivel al siguiente implica un salto de tensión de 5V/256, o unos 20mV.

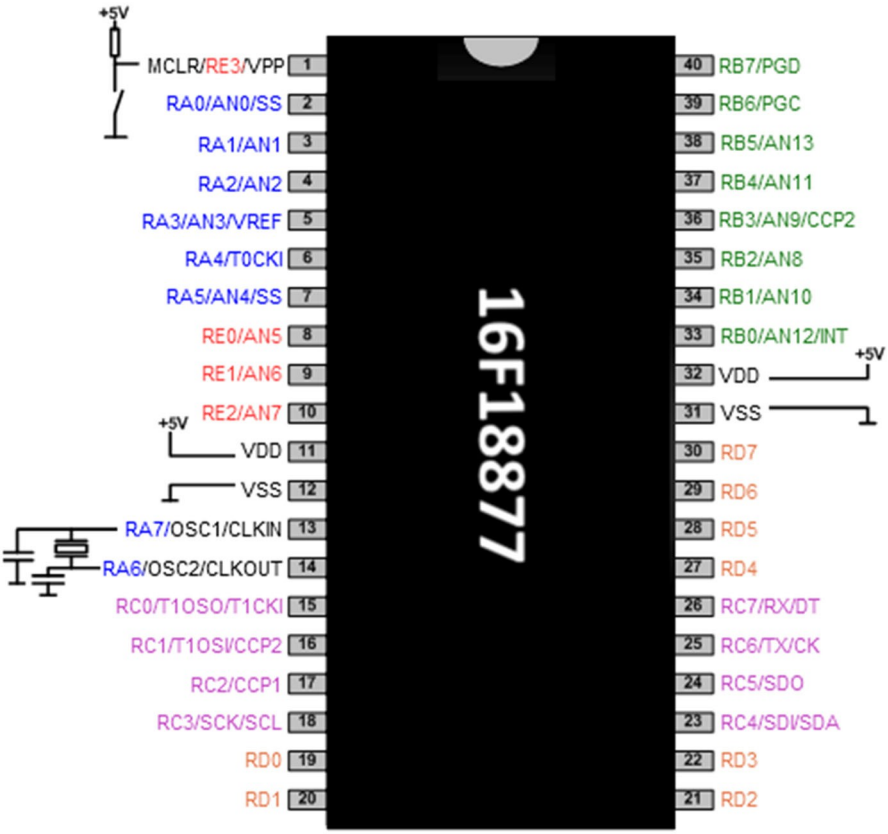
Cuando este microcontrolador procesa una señal analógica, primero la divide por 20 mV, para averiguar cuántos pasos incluye la señal. Así se obtiene el equivalente digital de la señal analógica.

El siguiente gráfico ilustra este proceso.



En nuestro ejemplo, el convertidor produce una salida '0000 0000'para cualquier señal analógica hasta 20mV, produce una salida '0000 0001'para señales analógicas entre 20 y 40mV, y así sucesivamente. La señal analógica mostrada en el gráfico produce una salida de "0000 0011".

El microcontrolador PIC es un dispositivo digital, pero los datos pueden introducirse tanto en forma analógica como digital. Los programadores eligen si los pines del PIC se utilizan como entradas analógicas, entradas digitales o salidas digitales. Esta flexibilidad da lugar a un etiquetado complejo.



El diagrama muestra el pinout de un chip PIC 16F18877. Tiene cinco puertos, conocidos como A, B, C, D y E. Los pines del puerto A están etiquetados de RA0 a RA7; los pines del puerto B están etiquetados de RB0 a RB7, etc. Los puertos A, B, C y D tienen ocho pines, pero el puerto E sólo tiene cuatro.

Por ejemplo, se pueden conectar hasta ocho sensores digitales al puerto A del 16F18877.

La clavija 2 está marcada como "**RA0/AN0**", lo que significa que que se puede utilizar como bit 0 del puerto A (Registro **A** bit 0) o como entrada **ANálogo** 0.

La función de cada pin de entrada/salida se determina ajustando el contenido de los registros internos, llamados registros de 'dirección de datos' dentro del dispositivo PIC.

Los pines RA6 y RA7 también se denominan "OSC1" y "OSC2". Pueden conectarse a un circuito oscilador externo o utilizarse como entrada/salida digital.

Los sensores analógicos deben conectarse a los pines etiquetados con 'ANx'(ANálogo). Estos, que se encuentran en los puertos A, B y E, pueden manejar señales analógicas entre V_{DD} (5V) y V_{SS} (Gnd).

La mayoría de los pines tienen funciones alternativas. Por ejemplo, el pin 25 está etiquetado como 'RC6/TX/CK', lo que significa que puede ser el bit 6 del Registro C, o el pin de transmisión (TX) de la interfaz serie interna, o el pin **ClocK** de la interfaz serie interna.

Afortunadamente Flowcode se encarga de los ajustes internos que dictan la funcionalidad de los pines por ti.

Salida de datos

El microcontrolador es un dispositivo digital, ya lo hemos dicho varias veces. Emite una señal digital. En la mayoría de los casos, la utilizamos para encender y apagar algo: "0"= "apagado" y "1"= "0n", por ejemplo.

Supongamos que configuramos el puerto B como puerto de salida, (o dejamos que Flowcode lo haga por nosotros). Hay ocho pines en el puerto B, por lo que podemos encender y apagar ocho dispositivos. Es importante planificar cómo conectamos estos dispositivos, ya que de lo contrario podrían funcionar al revés.

Salida de datos

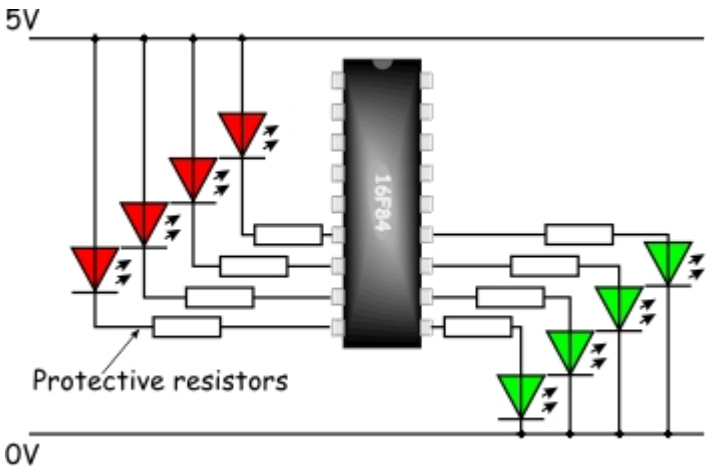
El diagrama muestra ocho LEDs conectados al puerto B de un microcontrolador PIC16F84:

Los cuatro LED rojos están conectados entre el carril positivo de alimentación y los pines del puerto B

Para estos LEDs, PIC es 'sinking'current.

Los cuatro LEDs verdes están conectados entre los pines y el rail de 0V.

Para ellos, PIC es 'sourcing'current.



Cada LED rojo se enciende cuando su pin está a baja tensión, es decir, cuando emite un "0".
Cada LED verde se enciende cuando su pin está a un voltaje alto, emitiendo un '1'.

(Hay límites en cuanto a la cantidad de corriente que los puertos pueden controlar. Típicamente, un pin de salida puede manejar hasta 25mA. Esto es suficiente para controlar LEDs y zumbadores directamente, pero los dispositivos de mayor potencia necesitarán circuitos adicionales para interactuar con el PIC, de los que hablaremos más adelante. Sin embargo, la corriente máxima para todo el puerto es de unos 100mA, por lo que no todos los pines pueden dar salida a 25mA al mismo tiempo).

Límites actuales

Como has visto, Flowcode tiene un modo de simulación que te permite conectar LEDs para mostrar el estado de los pines en el microcontrolador cuando se utilizan como salidas. La función de simulación de LEDs dentro de Flowcode asume que la corriente proviene del dispositivo PIC - como los LEDs verdes en el diagrama anterior.
En algún momento, tendrás que utilizar las especificaciones de los pines del PIC para utilizarlos como entradas digitales, entradas analógicas o como salidas digitales. En particular, existen limitaciones en las capacidades de salida del dispositivo. Exceder estos límites, aunque sea por poco tiempo, puede causar daños permanentes en el PIC.

Afortunadamente, todas las placas E-block utilizadas en este curso tienen resistencias limitadoras de corriente que protegen el dispositivo PIC. Sin embargo, cuando se utilizan las placas prototipo o patch, no existe tal protección y hay que tener cuidado de no dañar el dispositivo.

Corriente máxima absorbida/suministrada por cualquier pin de E/S	25 mA
Corriente máxima hundida por todos los puertos	200 mA
Corriente máxima suministrada por todos los puertos	140 mA
Corriente máxima de salida de la patilla VSS (Gnd)	95 mA
Corriente máxima en el pin V_{DD} (5V)	70 mA

Almacenamiento de datos

Los subsistemas electrónicos que almacenan datos se denominan "memoria". Sólo pueden almacenar datos digitales.

Un dato se almacena en una ubicación de la memoria. Estos datos pueden ser la combinación correcta para desactivar una alarma antirrobo o la temperatura objetivo del bloque motor de un coche.

Cada posición de memoria tiene una dirección única, un número que sirve para identificarla. Esto significa que podemos trazar un mapa de la memoria, mostrando qué datos contiene cada ubicación.

La versión decimal de la dirección se incluye para facilitar la lectura de la tabla.

Dirección		Datos almacenados
En decimal	En binario	
0	000	11101001
1	001	00100101
2	010	10000101
3	011	11001101
4	100	01110100
5	101	00011011
6	110	11110011
7	111	10000101

Los sistemas electrónicos sólo entienden números binarios. Esta pequeñísima memoria tiene ocho posiciones. (Necesita un número binario de 3 bits para crear direcciones únicas para cada ubicación. Permite almacenar datos de ocho bits de longitud (un "byte"(1B).

Nuestra memoria de ejemplo podría llamarse memoria 8 x 1B. Los sistemas de memoria utilizados en los ordenadores son mucho mayores. Los datos suelen almacenarse como números de 32 bits, lo que permite utilizar números mucho más grandes. También hay muchas más ubicaciones. Una memoria de ordenador típica tiene ahora millones de posiciones de memoria.

Tipos de memoria

Existen varios tipos de memoria electrónica, cada uno con una función ligeramente

diferente. Podemos dividirlas en dos grandes grupos: ROM y RAM:

Memoria de sólo lectura (ROM)

Normalmente, estos dispositivos sólo se leen (es decir, se accede a su contenido, pero no se modifica su "escritura") durante la ejecución de un programa.

- El contenido no es volátil. (Los datos permanecen almacenados aunque se desconecte la alimentación).
- Suelen utilizarse para almacenar los programas básicos, conocidos como "sistemas operativos", que necesitan los ordenadores.
- El grupo incluye:
    - PROM (memoria programable de sólo lectura),
    - EPROM (memoria de sólo lectura programable y borrable),
    - EEPROM (memoria de sólo lectura programable y borrable eléctricamente)

Una PROM es un dispositivo de un solo uso, que llega en blanco, listo para recibir datos. Los datos se pueden "grabar" en ella, pero sólo una vez. Después se comporta como un chip ROM que se puede leer muchas veces pero no alterar.

Con una EPROM, una luz ultravioleta a través de una ventana en la parte superior del chip borra el contenido. A continuación, se pueden "grabar" nuevos datos en la memoria. Algunos dispositivos PIC antiguos funcionan de este modo.

Los dispositivos EEPROM funcionan de forma similar a una EPROM, salvo que el contenido se borra enviando una secuencia especial de señales eléctricas a los pines seleccionados. La memoria Flash es una forma de EEPROM muy utilizada como medio de almacenamiento en cámaras digitales (el lápiz de memoria) y en consolas de videojuegos domésticas.

Memoria de acceso aleatorio (RAM)

- La memoria RAM permite tanto operaciones de lectura como de escritura durante la ejecución de un programa.
- Los contenidos son volátiles y desaparecen en cuanto se quita la alimentación. (La excepción es la NVRAM, Non -RAM volátil, donde el dispositivo de memoria puede incluir una batería para retener el contenido, o puede incluir un chip EEPROM como parte de la memoria para almacenar el contenido durante la pérdida de energía).
- Suelen utilizarse para el almacenamiento temporal de datos o programas de aplicación.

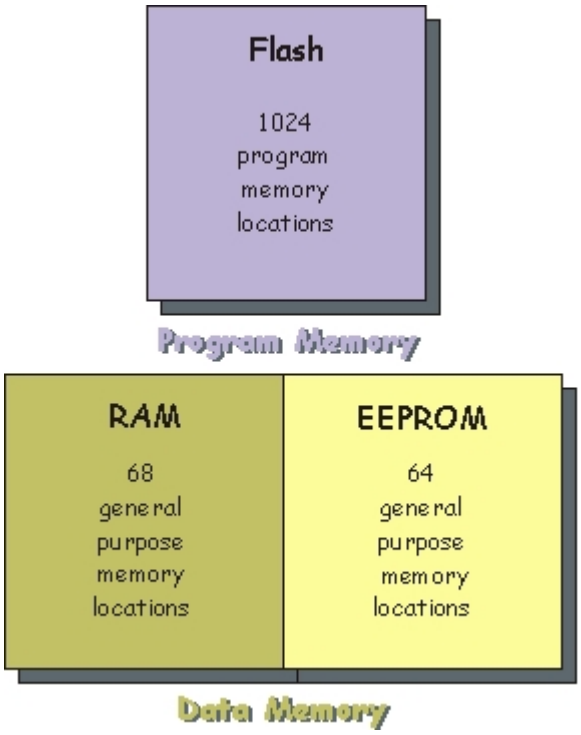
Memoria del microcontrolador

Los chips PIC tienen tres áreas de memoria separadas:

- memoria de programa (Flash);
- memoria variable de usuario (RAM);
- EEPROM.

Los nombres dan fuertes pistas sobre la finalidad de las zonas.

Para el PIC16F84 de dieciocho pines, el gráfico ilustra la organización de la memoria:



La memoria de programa se utiliza para almacenar el programa.

En la mayoría de los PIC, como el 16F18877, utiliza tecnología "Flash", lo que significa que puede programarse y borrarse muchas veces.
Los PIC más antiguos utilizan PROM para la memoria de programa, por lo que muchos de ellos sólo pueden programarse una vez.

La memoria de datos se utiliza para almacenar datos.

Parte de esto utiliza RAM y parte EEPROM.

La EEPROM permite conservar datos importantes aunque se desconecte la alimentación del sistema.

Por ejemplo, supongamos que el PIC forma parte de una temperatura que mantiene una incubadora a una temperatura determinada. Podría tener sentido almacenar el valor de temperatura objetivo en EEPROM para no tener que introducirlo en el sistema cada vez que encendamos la incubadora.

Programación

Los microcontroladores son dispositivos programables. Hacen exactamente lo que les dice el programa, ¡y nada más! Un programa es una lista de instrucciones, junto con los datos necesarios para llevarlas a cabo.

Lo único que entienden los microcontroladores son números. ¡Hay un problema! Nosotros no hablamos en números, ¡y ellos no entienden inglés!

Hay dos soluciones, y ambas necesitan algún tipo de traductor:

- Escribe el programa en inglés, o algo parecido, y luego traduce el resultado a números.
- Podemos pensar el diseño del programa en inglés y luego traducirlo nosotros mismos a un lenguaje similar a los números, conocido como "ensamblador". A partir de ahí, es un paso rápido y sencillo convertirlo en el código numérico que entiende el microcontrolador.

Estos dos extremos se conocen como programación en un **lenguaje de alto nivel** (algo parecido al inglés) o en un **lenguaje de bajo nivel** (ensamblador).

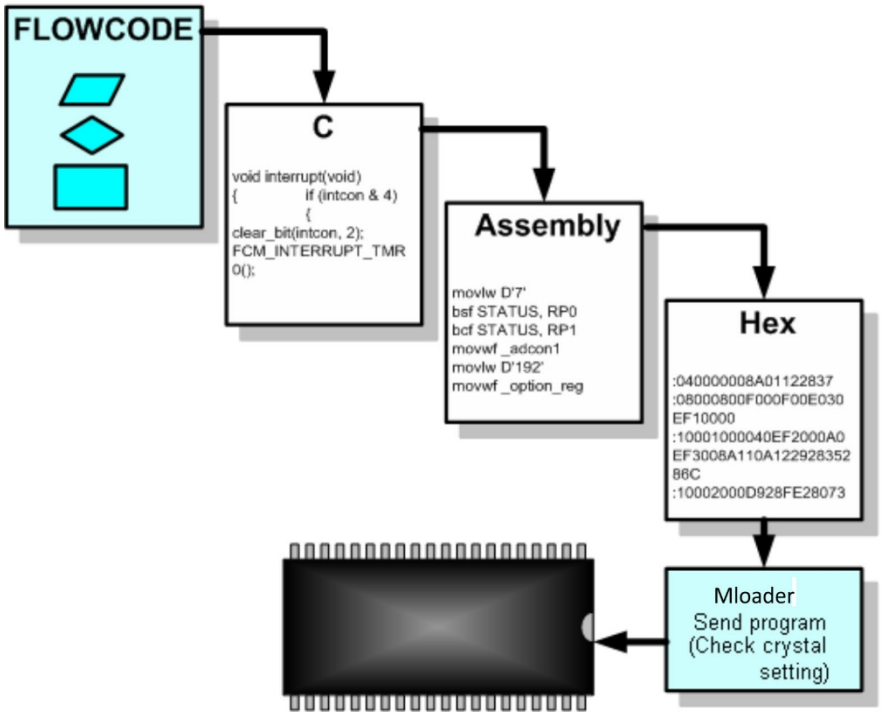
La primera suele ser más rápida y sencilla para el programador, pero tarda más en ejecutar el programa, debido a la necesidad de traducirlo para el microcontrolador.
La segunda es mucho más lenta para el programador, pero acaba ejecutándose muy rápidamente en el microcontrolador.
Si cree que esto suena muy complicado, tiene razón. Lo es. Afortunadamente, Flowcode trabaja utilizando diagramas de flujo - el más fácil, y el más alto nivel, de programación y luego se encarga de toda la traducción necesaria.

El proceso Flowcode

'Flowcode ofrece una forma sencilla de programar chips microcontroladores, como verás. Una vez diseñado el diagrama de flujo en pantalla, basta con pulsar un botón para que el software lo traduzca en código numérico.

Flowcode hace pasar el programa por una serie de procesos antes de enviarlo al microcontrolador.
Se procesa el diagrama de flujo:

- primero en código C,
- y luego en Assembler,
- y finalmente en números hexadecimales o 'Hex', que el microcontrolador 'entiende'.



A continuación, el código hexadecimal se envía al microcontrolador mediante un programa subsidiario llamado "Mloader".
Al seleccionar *Build > Project Options... Configurar* en el menú Flowcode, se ejecuta el programa 'Mloader'. Controla una serie de opciones y configuraciones estableciendo el valor de los registros dentro del dispositivo cuando descarga un programa.

El código hexadecimal se "graba" en la memoria de programa del microcontrolador. Dado que la memoria Flash se utiliza para formar la memoria de programa, el programa no se pierde cuando se retira el microcontrolador del programador. Esto permite utilizarlo en un circuito. Igualmente, el uso de memoria Flash significa que puedes reutilizar el microcontrolador y sobrescribir la memoria de programa con un nuevo programa.

Ejecución del programa

Tan pronto como el microcontrolador se enciende y se alimenta con pulsos de reloj, comenzará a ejecutar cualquier programa que esté almacenado en la memoria de programa (Flash).

Al pulsar el botón de reinicio de la placa de programación del microcontrolador, el programa se reinicia desde el principio.

Durante la programación, el microcontrolador se detiene mientras se carga el programa. Cuando finaliza, se reinicia y ejecuta el programa descargado.

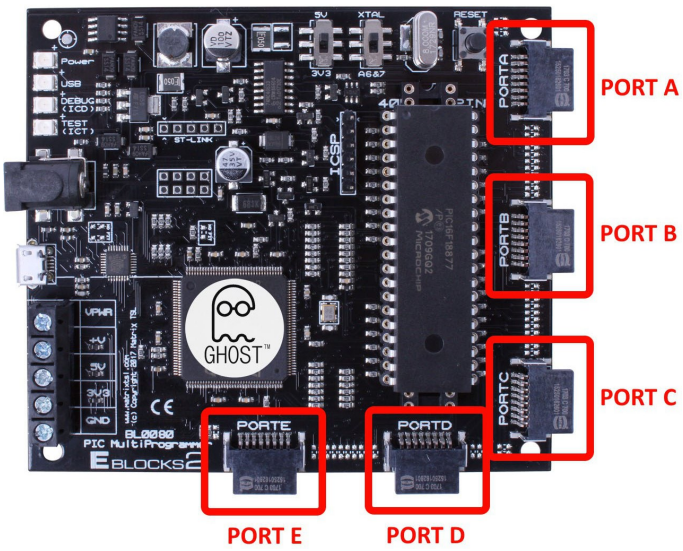
Diferentes tipos de microcontroladores

Hay un gran número de microcontroladores disponibles, desde el humilde 16F84 hasta microcontroladores más grandes y complejos, como el 16F18877 de 40 pines. Los distintos microcontroladores tienen diferente número de puertos, o pines de E/S, entradas analógicas, mayor memoria o capacidades avanzadas de comunicación serie, como RS232 o bus SPI.

Decidir qué dispositivo utilizar para un proyecto puede ser una tarea en sí misma. Para este curso usamos un dispositivo 16F18877, un PIC de 40 pines que tiene muchos subsistemas internos (como un convertidor A/D, y un puerto serie).

Arquitectura del PIC16F18877

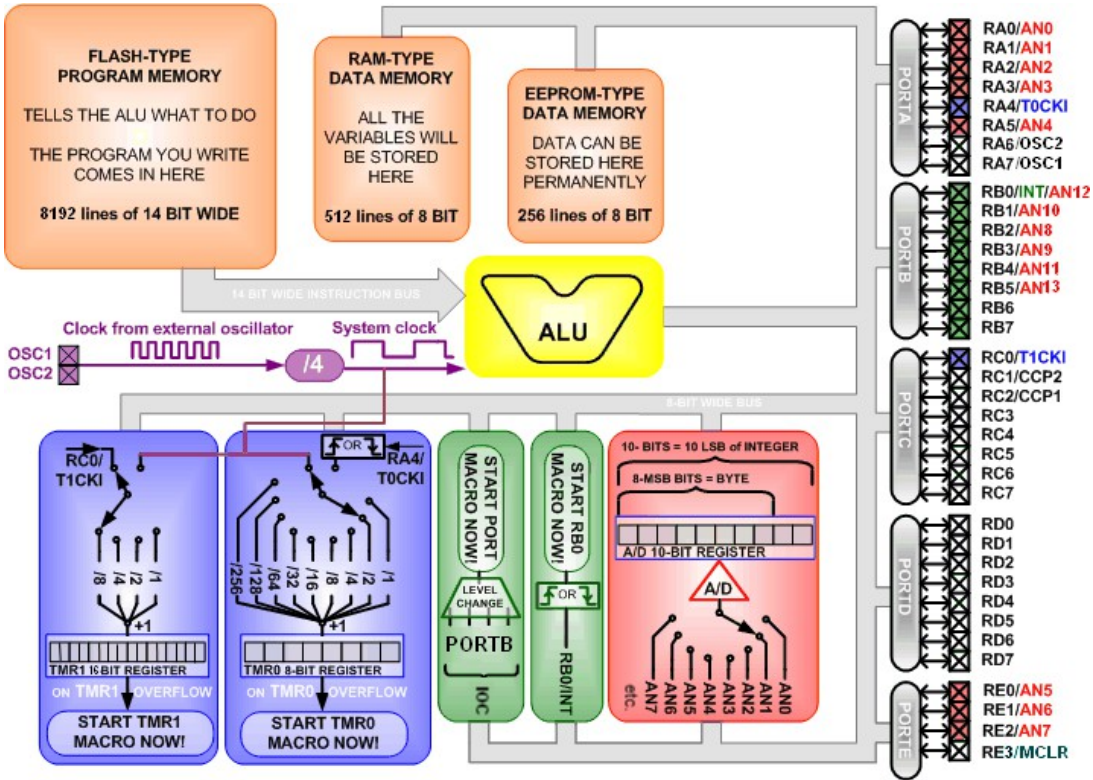
Como este curso utiliza el PIC16F18877, es importante que entiendas un poco más sobre lo que hace y cómo utilizarlo. Esta sección detalla los pines que están disponibles en el 16F18877 y los conectores que utilizan en la placa programadora. (La sección "Usando E-blocks" explica como se hacen estas conexiones).



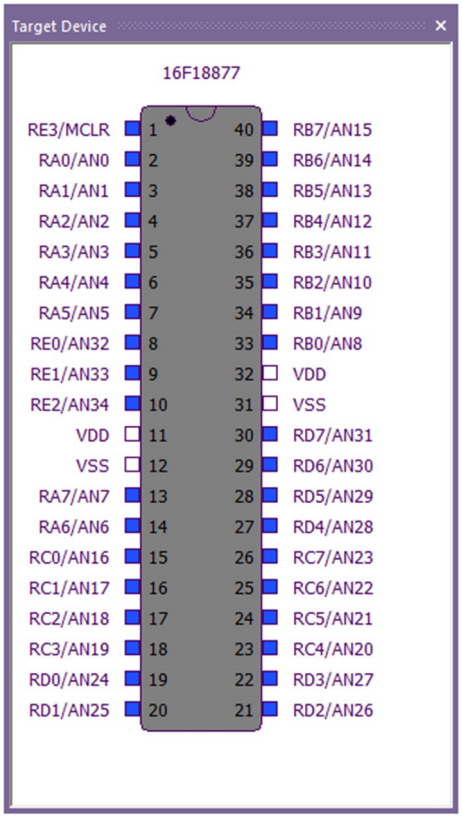
En este punto, en un curso de programación tradicional, se te presentarían con cierto detalle los distintos bloques de circuitos internos del dispositivo PIC. Necesitarías esta información para escribir código para el PIC en C o en ensamblador. No es necesario - ¡Flowcode se encarga de estos detalles!

Sin embargo, es necesario conocer las conexiones de entrada y salida del PIC, la memoria disponible y la función de los demás subsistemas del PIC.

Puertos - El PIC16F18877 PIC tiene cinco puertos, etiquetados de 'A'a 'E', conectados al resto de las partes internas del microcontrolador mediante un sistema de bus de 8 bits.



La distribución de pines del PIC16F8877:



Otros subsistemas en el PIC16F18877:

Memoria:

Flash

- La memoria Flash se utiliza para almacenar el programa que escribes.
- El ordenador 'compila' este programa en código binario y lo descarga en la memoria Flash del PIC.
- Se puede leer y escribir en él, y se conserva incluso después de un corte de luz.
- La memoria Flash contenida en el 16F18877 puede almacenar hasta 32768 comandos de programa.

RAM

- Los datos de entradas, salidas, entradas analógicas, cálculos, etc. suelen almacenarse en "variables" (valores del programa que se modifican a medida que éste se ejecuta). La memoria RAM es donde se almacenan.
- Esta memoria se borra cada vez que se corta la corriente o se produce un reinicio.
- También contiene "registros" del sistema que controlan e informan del estado del dispositivo.
- La memoria RAM del 16F18877 puede almacenar hasta 4096 bytes de datos.

EEPROM

- EEPROM es donde los datos pueden ser almacenados permanentemente
- Esta memoria es del tipo PROM - se conserva cada vez que se corta la corriente o se produce un reinicio.
- La EEPROM del 16F18877 puede almacenar hasta 256 bytes de datos.

ALU:

- La ALU (Unidad Aritmética Lógica) es el corazón del procesamiento de datos del PIC.
- Todos los datos pasan por esta unidad.
- El programa de la memoria Flash indica a la ALU lo que debe hacer.
- La ALU puede enviar y recuperar datos de todos los bloques y puertos del PIC utilizando el bus de datos de 8 bits de ancho.
- La ALU necesita cuatro impulsos de reloj del oscilador externo para ejecutar una instrucción completa.
- El funcionamiento de la ALU es muy complicado. Afortunadamente los programadores de Flowcode no necesitan saber cómo funciona.

Temporizador 1 (TMR1):

- Esta interrupción del temporizador se utiliza para proporcionar al microcontrolador información exacta sobre la temporización.
- Se sincroniza con el reloj del sistema o con un reloj externo seleccionable en los pines A0 a A7 o C0-C7.
- Cualquiera de los dos relojes puede dividirse por 1, 2, 4 u 8 configurando el Prescaler de TMR1 en Flowcode. La salida resultante activa TMR1 e incrementa el registro TMR1.
- TMR1 es un registro de 16 bits que se 'desborda' cuando alcanza '65536'.
- En el momento en que se desborda, genera una interrupción y el registro TMR1 se pone a '0'.
- Esta Interrupción TMR1 detiene el programa principal inmediatamente y lo hace saltar a la macro TMR1.
- Después de esto, el programa principal continúa desde donde lo dejó justo antes de la interrupción.

Por ejemplo:

Frecuencia del oscilador de reloj externo (oscilador de cristal)	32 MHz
Reloj del sistema (cuatro impulsos de reloj por instrucción)	8 MHz
Ajuste el preescalador a '8'(divide por 8)	1MHz
Frecuencia de desbordamiento cuando TMR1 = '65536'	15,259 Hz

Resultado:

TMR1 interrumpe el programa principal y ejecuta la macro TMR1 15,259 veces por segundo.

Temporizador 0 (TMR0):

- Esta interrupción del temporizador también proporciona al microcontrolador información exacta sobre la temporización.
- Se sincroniza con el reloj del sistema o con un reloj externo seleccionable en los pines A0 a A7 o B0-B7.
- Este reloj del sistema funciona exactamente cuatro veces más lento que el reloj del oscilador externo.
- Cualquier reloj se puede dividir por 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 161384 y 32768 configurando el Prescaler de TMR0 en Flowcode. El resultado dispara TMR0 e incrementa el registro TMR0.
- Se pueden hacer más divisiones seleccionando un valor de postescalado de 1:1 a 1:16
- Este registro TMR0 es un registro de 16 bits, que se desborda cuando llega a 65535.
- En el momento en que se desborda, genera una interrupción y el registro TMR0 se pone a 0.
- Una interrupción TMR0 detiene el programa principal inmediatamente y lo hace saltar a la macro TMR0. Después de que esto termine, el programa principal continúa desde donde lo dejó justo antes de la interrupción.

Por ejemplo:

Frecuencia del oscilador de reloj interno (oscilador de cristal)	32 MHz
Reloj del sistema (4 impulsos de reloj por instrucción)	8 MHz
Ajuste el preescalador a 1 (divide por 65536 y luego por 1)	122,07 Hz
Poner el Postscaler a 1 (divide por 1)	122,07 Hz

Resultado: TMR0 interrumpe el programa principal y ejecuta la macro TMR0 122 veces por segundo. Para una mayor precisión, utilice el temporizador 2, ya que se pueden ajustar los valores de preescalado, posescalado y rollover.

Interrupción externa RB0:

- Un cambio de nivel lógico en el pin RB0 puede configurarse para generar una interrupción.
- Se puede configurar en Flowcode para que reaccione a un flanco ascendente o descendente en RB0.
- Si se configura para reaccionar a un flanco ascendente, cuando se produce uno:
    - detiene inmediatamente el programa principal;
    - se ejecuta la macro relacionada con RB0;
    - entonces el programa principal continúa desde donde lo dejó justo antes de la interrupción.

Esto ocurre cada vez que se detecta un flanco ascendente en el pin RB0.

Interrupción externa del PUERTO B:

- Un cambio de nivel lógico en cualquier combinación de pines del puerto B puede generar una interrupción.
- Puede configurarse para que se produzca en un flanco ascendente o descendente, o en ambos.
- Cuando se produce una de estas interrupciones:
    - detiene inmediatamente el programa principal;
    - se ejecuta la macro relacionada con el puerto B;
    - entonces el programa principal continúa desde donde lo dejó justo antes de la interrupción.

Esto ocurre cada vez que se detecta un cambio de nivel en uno de los pines seleccionados en el puerto B.

A/D:

- El 16F18877 tiene treinta y cinco pines que tienen una función A/D extra.
- Sólo tiene un convertidor A/D de 10 bits.
- Esto implica que estas treinta y cinco entradas analógicas no pueden leerse todas al mismo tiempo.
- Un conmutador analógico integrado, configurado en Flowcode, selecciona qué entradas se muestrean.
- Tras la instrucción 'sample', el conmutador analógico apunta a la entrada correcta y ésta se convierte en un valor binario de 10 bits.
- En Flowcode, puede optar por utilizar sólo los ocho bits más significativos (MSB) de este valor de 10 bits, mediante la instrucción 'GetByte', o utilizar los diez bits completos mediante la instrucción 'GetInt'. Los diez bits llenarán los diez bits menos significativos (LSB) de la variable entera de 16 bits seleccionada.
- Después de esto, el programa puede seleccionar leer otra entrada analógica.

Autobuses:

- Los microcontroladores PIC y AVR (Arduino) utilizan la arquitectura Harvard.
- Esto significa que hay buses separados para las instrucciones y para los datos.
- El bus de datos tiene 8 bits de ancho y conecta todos los bloques y puertos entre sí.
- El bus de instrucciones tiene 14 bits de ancho y transporta las instrucciones, que tienen 14 bits de longitud, desde la memoria de programa hasta la ALU.

Introducción a los "relojes

Todo microcontrolador necesita una señal de reloj para funcionar. Internamente, la señal de reloj controla la velocidad de funcionamiento y sincroniza el funcionamiento de los distintos bloques de hardware internos.

En general, los microcontroladores se pueden 'cronometrar' de varias maneras, utilizando:

- un oscilador de cristal externo;
- Modo 'RC', en el que la frecuencia del reloj depende de una resistencia y un condensador externos;
- un oscilador interno.

El modo "RC" es en parte histórico y en parte económico. Se introdujo como una alternativa de bajo coste a un oscilador de cristal. Está bien para aplicaciones que no son críticas para la temporización, pero no se trata en este curso.

Sección 2: Utilizando Bloques E

Los E-Blocks son pequeñas placas de circuitos que pueden conectarse fácilmente entre sí para formar un sistema electrónico. Hay dos tipos de E-Blocks. Placas **ascendentes** y placas **descendentes**.

Se pueden combinar varias placas para crear un sistema completo con placas conectadas a placas anteriores.

Los E-blocks son los compañeros ideales del software Flowcode, ya que permiten a los usuarios probar y desarrollar sus programas Flowcode. Los programas se pueden compilar directamente en las placas, lo que proporciona entornos de desarrollo ideales.

Los bloques E constan de placas ascendentes y placas descendentes.

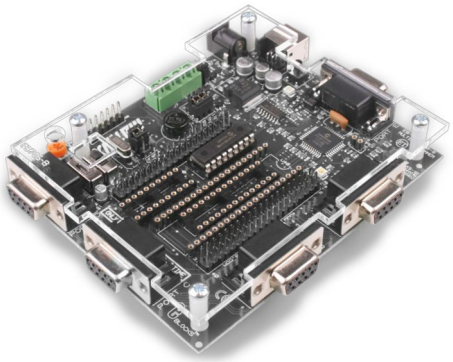
Tableros ascendentes

'Upstream' es un término informático que indica una placa que controla el flujo de información en un sistema. Suelen estar programadas de alguna manera.

Cualquier dispositivo que contenga "inteligencia" y pueda dictar la dirección del flujo de información en el bus puede considerarse un dispositivo "ascendente".

Algunos ejemplos son las placas de microcontroladores y las placas de dispositivos lógicos programables.

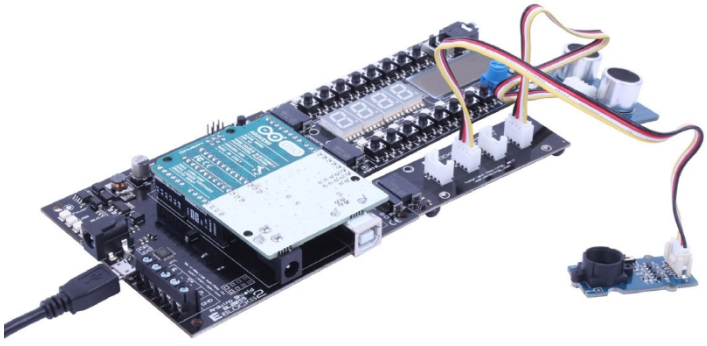




Tableros descendentes

Las tarjetas "aguas abajo" están controladas por una tarjeta "aguas arriba", pero la información puede entrar o salir de ellas. Algunos ejemplos son las tarjetas LED, LCD, RS232, etc.

Las placas aguas arriba y aguas abajo se combinan para formar un sistema completo, en el que las placas aguas abajo se conectan a las placas "inteligentes" aguas arriba:



BL0011 Programador PIC

- La placa tiene cinco puertos, etiquetados de A a E.
- Los puertos "B", "C" y "D" ofrecen funcionalidad completa de 8 bits.
- El puerto 'A' tiene una funcionalidad de 6 bits (8 bits si se selecciona el oscilador interno).
- El puerto 'E' tiene una funcionalidad de 3 bits.
- Puede alimentarse desde una fuente de alimentación externa de 7,5 V a 9 V o desde una fuente USB.
- Si se pulsa el interruptor Reset, se reiniciará el programa almacenado en el microcontrolador.
- La placa es programable por USB a través de un chip de programación. Este se encarga de la comunicación entre Flowcode y el microcontrolador.
- El microcontrolador ejecuta una instrucción por cada cuatro impulsos de reloj que recibe.
- (Nota - una sola instrucción NO es lo mismo que un solo símbolo Flowcode, que se compila en C y luego en ensamblador y probablemente da como resultado una serie de instrucciones).
- Este curso utiliza un cristal de 8MHz que se multiplica hasta 32Mhz internamente.
- Los interruptores permiten al usuario seleccionar una serie de opciones.
- Fuente de alimentación externa o USB.
- Si el microcontrolador utiliza un oscilador interno, los ocho bits del puerto A pueden utilizarse para operaciones de E/S.
- Uso de una herramienta PICKit3 de Microchip a través de la cabecera ICSP.
- Incluye un dispositivo PIC16F18877 montado en superficie.
- Proporciona alimentación a las tarjetas E-blocks posteriores a través de los conectores de puerto.
- Contiene el chip Matrix Ghost que permite la depuración en circuito en tiempo real cuando se combina con Flowcode.

Para obtener información general sobre el programador Arduino, consulte el Apéndice 1, SECCIÓN A (página 89).

BL0114 Placa combinada

La placa combina en una sola placa compacta la funcionalidad que se encuentra en varias placas E-blocks individuales:

- BL0167 Placa LED (x2)
- BL0169 Placa LCD
- BL0145 Placa de interruptores (x2)

Para este curso, los conectores de los puertos se acoplan a los conectores hembra de los puertos A y B de la placa de subida. La placa proporciona un conjunto de ocho interruptores y ocho LED para el puerto A y lo mismo para el puerto B.

Con el interruptor principal en la posición DIG, el puerto A se dirige a sus interruptores pulsadores (SA0 a SA7), a los LEDs (LA0 a LA7) y al display cuádruple de 7 segmentos.

Con el interruptor principal en la posición ANA, el puerto A se conecta a la sección de sensores analógicos de la placa, de forma que el pin RA0 se conecta al sensor de luz de a bordo y el pin RA1 se conecta al potenciómetro para dar una tensión de salida variable, (simulando la acción de un subsistema sensor analógico).

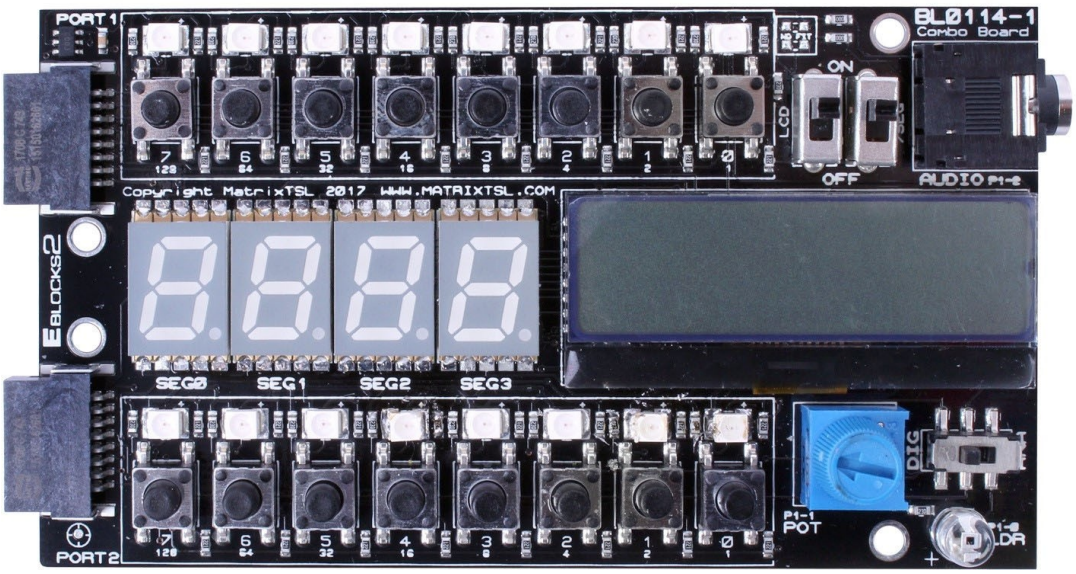
Nota: Con el interruptor en la posición ANA, los interruptores de a bordo y los LEDs LA0 y LA1 no funcionarán.

Los pines de E/S del puerto B se dirigen a sus interruptores pulsadores (SB0 a SB7), a los LEDs (LB0 a LB7), a los displays cuádruples de 7 segmentos y al display LCD.

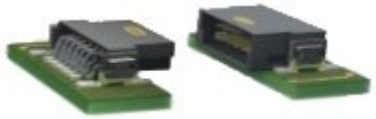
La pantalla cuádruple de 7 segmentos se enciende mediante el interruptor '7SEG'. Está conectado a los puertos A y B.

- El puerto B se utiliza para controlar los segmentos LED y el punto decimal).
- Puerto A, bits 0 a 3, seleccionan qué pantalla se activa.

El LCD es un módulo de 20 caracteres x 4 líneas, que se enciende mediante el interruptor 'LCD'. Normalmente es un dispositivo complejo de programar, Flowcode se encarga de las complejidades, invisibles para el usuario.

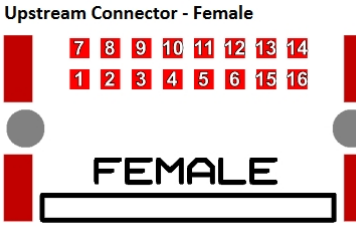


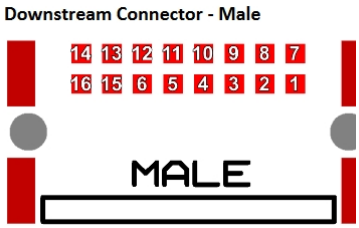
Conexión de bloques E

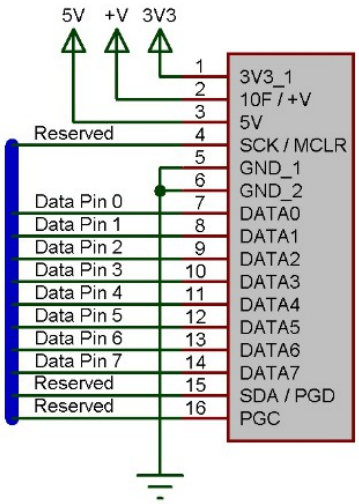


Los E-blocks2 se basan en un concepto de bus. Cada bloque E se conecta mediante un conector Har-flex de 16 clavijas; los puertos hembra se conectan a las placas "inteligentes" y los puertos macho a las placas "inteligentes". conectores acoplados a las placas de aguas abajo.

El diagrama muestra que los tres primeros pines se utilizan para transferir la alimentación a la placa aguas abajo, los pines 4,15 y 16 están reservados.

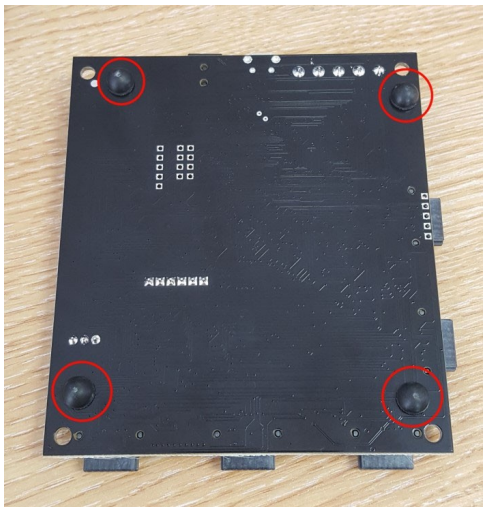






Los pines 5 y 6 están conectados a tierra, mientras que los pines 7 -14 son los pines que transfieren nuestros 8 bits de datos entre las placas.

Utilización de bloques E en el banco



Para utilizar los E-blocks no hace falta una placa base: basta con conectarlos entre sí en el banco. En cada paquete de E-blocks encontrará cuatro pequeños pies de goma para facilitar esta operación. Proporcionan cierta protección a las placas E-blocks y ayudan a evitar cortocircuitos con el cable de cobre estañado y otros objetos metálicos del banco. La desventaja es que el sistema E-blocks es menos portátil, ya que los conectores estarán sometidos a más tensión cuando se desplace el sistema.

Protección de los circuitos E-blocks

Siempre que ha sido posible, se han utilizado componentes con plomo para los dispositivos de las placas de bloques E susceptibles de sufrir daños eléctricos. Esto simplifica la tarea de sustituirlos en caso de que resulten dañados.

Para proteger los componentes "aguas arriba", todas las placas E-blocks "aguas abajo" incluyen resistencias de protección. Si se producen errores al declarar la naturaleza de los pines de puerto, por ejemplo, una entrada declarada como salida, no se producirá ningún daño.

Sin embargo, hay circunstancias en las que es posible causar daños:

- Hay que tener cuidado al utilizar conectores de terminales de tornillo y placas de parcheo/prototipos.
- Siempre que sea posible, utilice resistencias de protección para las líneas que tenga que conectar cuando conecte dos placas "aguas arriba" con un bloque E de cambio de género.
- Asegúrese de estar conectado a tierra antes de manipular las placas de circuitos de E-blocks para minimizar el riesgo de daños por electricidad estática. Si no dispone de una muñequera antiestática, toque un radiador u otro objeto metálico conectado a tierra.

Antes de realizar cualquier cambio en el sistema E-blocks, desconecte la fuente de alimentación.

Sección 3:

Introducción a la Flowcode Embedded

Flowcode Embedded le permite crear aplicaciones de microcontrolador arrastrando y soltando iconos en un diagrama de flujo para crear programas. Éstos pueden controlar dispositivos externos conectados al microcontrolador, como LED, pantallas LCD, etc.

Una vez diseñado el diagrama de flujo, puede simularse su comportamiento en Flowcode antes de compilarlo, ensamblarlo y transferirlo a un microcontrolador.

Esta sección permite a aquellos que son nuevos en Flowcode entender cómo se puede utilizar para desarrollar programas. Permite introducir programas paso a paso para aprender cómo funciona Flowcode.

Le aconsejamos que trabaje a través de cada sección para familiarizarse con todas las opciones y características de Flowcode e introducirle en una serie de técnicas de programación. A medida que avance en cada parte, consulte también el archivo de ayuda de Flowcode. A continuación se presentan los principales iconos de Flowcode.

Específicamente en esta sección aprenderás:

- cómo utilizar cada icono de Flowcode (excepto el icono de código C);
- cómo funcionan los componentes fundamentales de Flowcode: el LED, el LCD, el ADC, el interruptor, la pantalla de 7 segmentos, la pantalla cuádruple de 7 segmentos, el teclado y los componentes EEPROM.

¿Qué es Flowcode Embedded?

Flowcode Embedded le permite crear aplicaciones de microcontrolador arrastrando y soltando iconos en un diagrama de flujo para crear programas. Éstos pueden controlar dispositivos externos conectados al microcontrolador, como LED, pantallas LCD, etc.

Una vez diseñado el diagrama de flujo, puede simularse su comportamiento en Flowcode antes de compilarlo, ensamblarlo y transferirlo a un microcontrolador.

El proceso:

1. Cree un nuevo diagrama de flujo, especificando el microcontrolador al que desea apuntar.
2. Arrastre y suelte los iconos de la barra de herramientas en el diagrama de flujo para programar la aplicación.
3. Añada dispositivos externos haciendo clic en los botones de la barra de herramientas de componentes.
4. Edita sus propiedades, incluyendo cómo se conectan al microcontrolador, y configura cualquier macro que utilicen.
5. Ejecute la simulación para comprobar que la aplicación se comporta como se espera.
6. Transfiera la aplicación al microcontrolador compilando el diagrama de flujo a C, luego a código ensamblador y finalmente a código objeto.

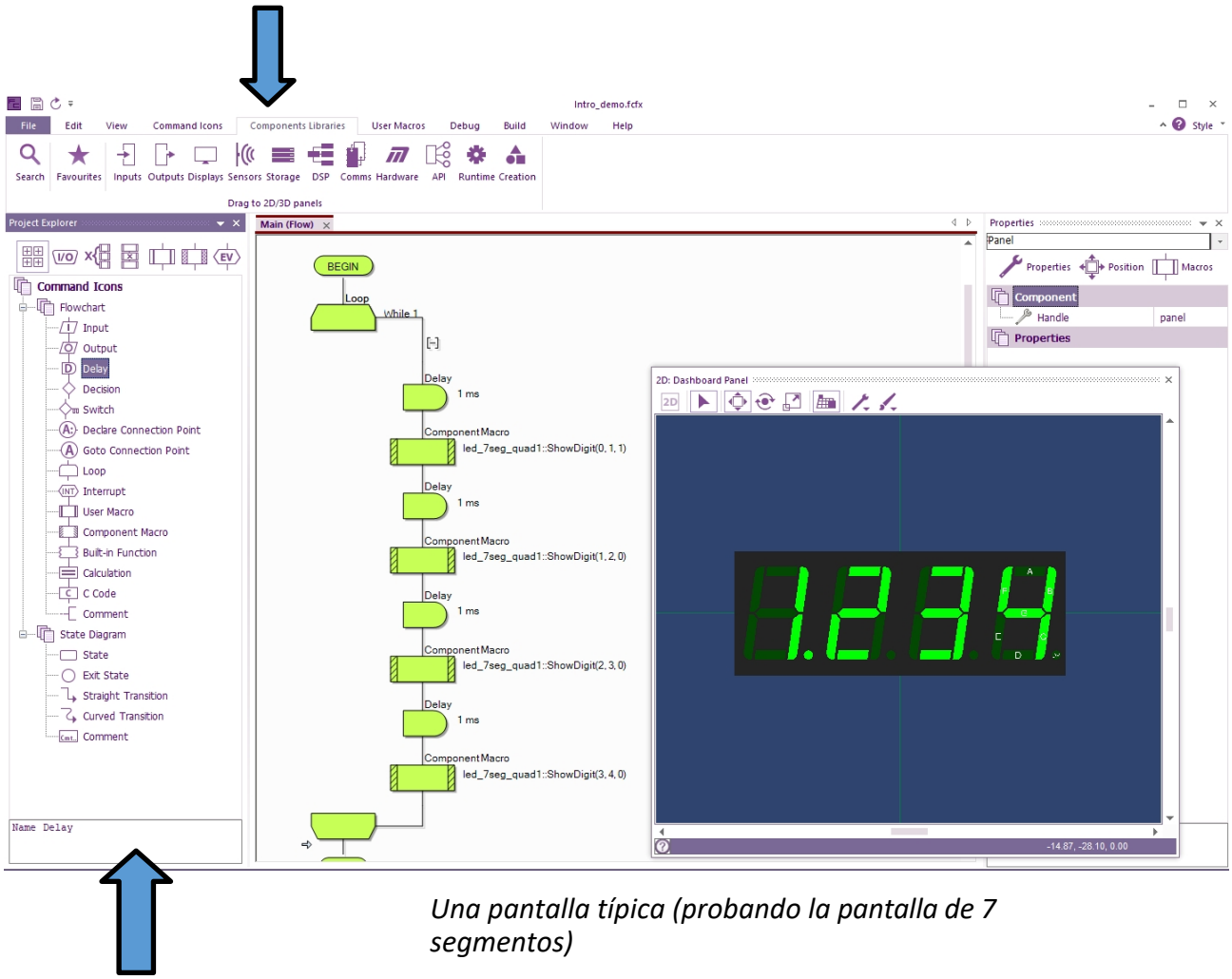
Visión general de Flowcode Embedded

El entorno Flowcode consta de:

- un área de trabajo principal en la que se muestran las ventanas del organigrama.
- una serie de barras de herramientas que permiten añadir iconos y componentes al organigrama.
- los paneles Sistema y Cuadro de mandos que muestran los componentes adjuntos y proporcionan información básica
- capacidades de dibujo.
- el panel del Explorador de proyectos que muestra las variables del proyecto, las macros y las macros de los componentes.
- el panel Lista de iconos que muestra marcadores, puntos de interrupción y resultados de búsqueda.
- ventanas que permiten ver el estado del microcontrolador.
- ventanas que muestran variables y llamadas a macros cuando se simula el diagrama de flujo.

Barra de herramientas de las bibliotecas de componentes

Conecte componentes externos al microcontrolador o utilice comandos básicos de dibujo de paneles. Los componentes están agrupados en diferentes categorías que aparecen como menús desplegables. Haga clic en un componente y éste se añadirá al microcontrolador y aparecerá en el panel. A continuación, se pueden editar las conexiones de patillas y las propiedades del componente.

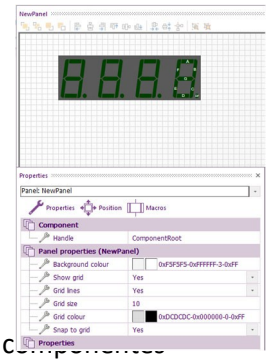


Una pantalla típica (probando la pantalla de 7 segmentos)

Iconos de mando

Arrastre y suelte los iconos de esta ventana en la ventana principal del diagrama de flujo para crear la aplicación del diagrama de flujo. Alternativamente, los iconos están disponibles en la barra de herramientas de iconos de comandos. Esta es la primera pestaña del Explorador de Proyectos, aquí acoplada a la izquierda, pero puede desacoplarse.

Paneles 2D, paneles de sistemas 2D y 3D heredados



Los componentes que conectes al microcontrolador se mostrarán en uno de estos paneles, que también ofrecen funciones básicas de dibujo, como líneas, formas e imágenes.

El Panel 2D es principalmente para uso 2D.

Puede añadir tantos paneles 2D como desee, todos ellos con nombres individuales. La apariencia se puede cambiar dentro de las propiedades del panel.

Generalmente se utilizan como interfaz donde se guardan los botones e interruptores de los componentes interactivos. El panel de sistema es el panel 3D

Encontrará más detalles sobre estos paneles en 'Flowcode - Guía de introducción'.

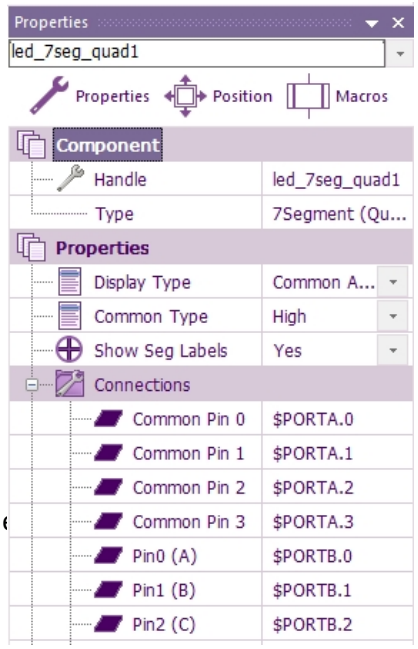
(Ver > Paneles 2D > Añadir nuevo panel 2D...) / (Ver > Panel 2D heredado) / (Ver > 3D: Panel del sistema)

Panel de propiedades de los componentes

Todos los elementos del panel, incluido el propio panel, tienen propiedades asociadas que se muestran en el panel de propiedades cuando se selecciona el elemento.

Algunos son de sólo lectura, mientras que otros pueden manipularse.

Algunos, como el tamaño y la posición, cambian al interactuar con el elemento. Otros permiten acceder a funciones más avanzadas del elemento seleccionado.



El panel de propiedades suele estar anclado a la derecha de la pantalla, pero tiene el siguiente aspecto cuando está desanclado: **(Ver > Propiedades del componente)**

Explorador de proyectos

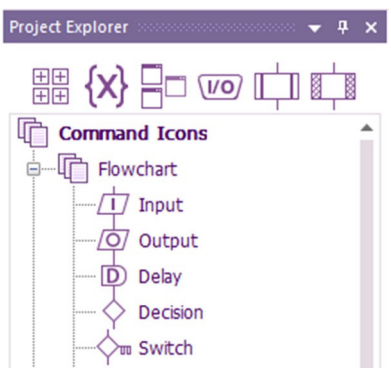
Los botones de la parte superior de este panel permiten seleccionar "Puertos", "Globales", "Macros" y "Componentes".

La vista 'Puertos' muestra los nombres de las variables asignadas a los puertos de

La vista 'Globales' muestra todas las constantes y variables que se han definido para su uso en el proyecto actual.

La vista "Macros" muestra las macros creadas por el usuario en el programa actual y le permite arrastrarlas al diagrama de flujo actual.

La vista "Componentes" es muy similar, salvo que también enumera los componentes presentes en el panel. **(Vista > Explorador de proyectos)**

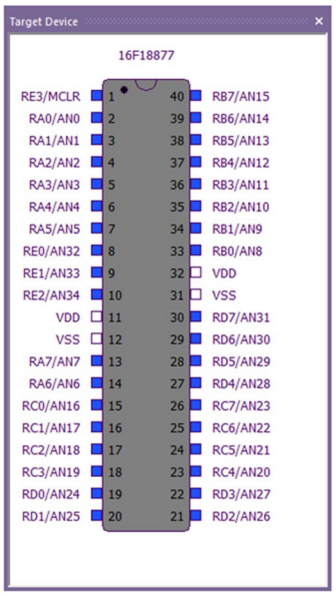


Ventana del dispositivo de destino

Se muestra el pinout del chip microcontrolador actualmente seleccionado.

Cuando se simula el diagrama de flujo, el estado de los puertos de E/S del microcontrolador se muestran en rojo y azul, para salidas altas y bajas respectivamente.

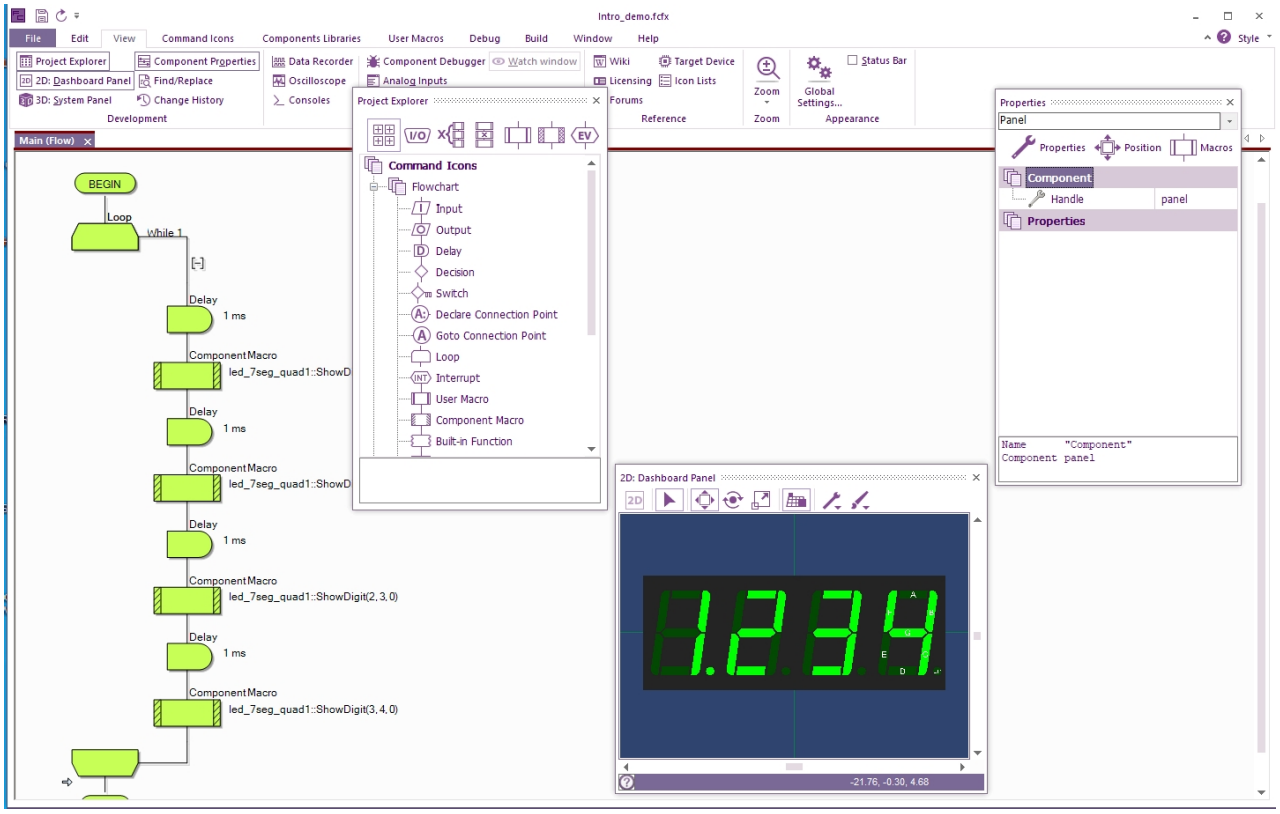
- **(Ver > Dispositivo de destino)**



Acoplar y desacoplar las barras de herramientas y los paneles

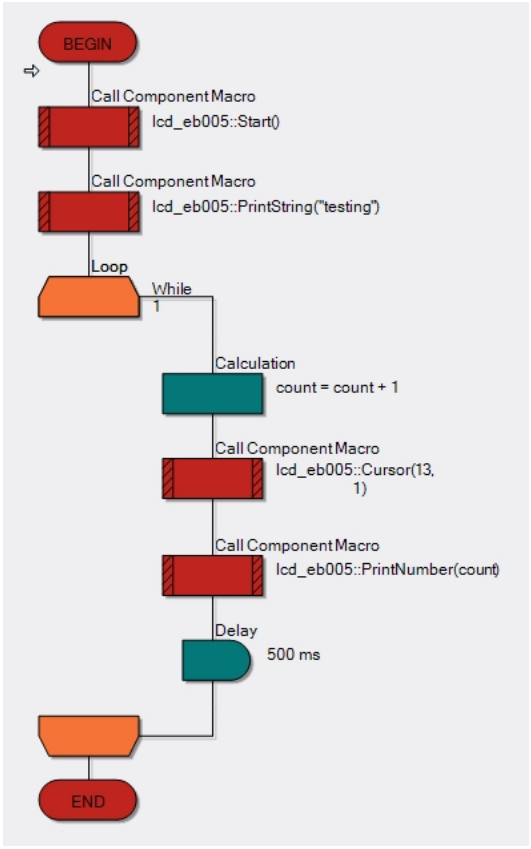
Las barras de herramientas y los paneles pueden desacoplarse de sus posiciones predeterminadas y dejarse flotando libremente, o acoplarse a la parte superior, inferior o a los lados de la ventana de Flowcode.

Un ejemplo de barras de herramientas flotantes:

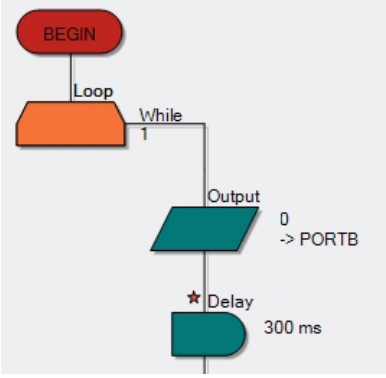


Ventana de diagrama de flujo

En este espacio principal se muestran los iconos que componen el diagrama de flujo. El texto cambiará en función de las propiedades seleccionadas, las macros de componentes activadas, etc. El usuario puede cambiar los nombres de visualización para facilitar la organización del proyecto.

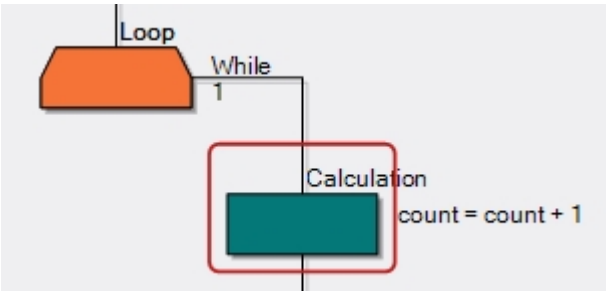


Una estrella roja junto a un icono indica que el organigrama no se ha guardado en su forma actual.



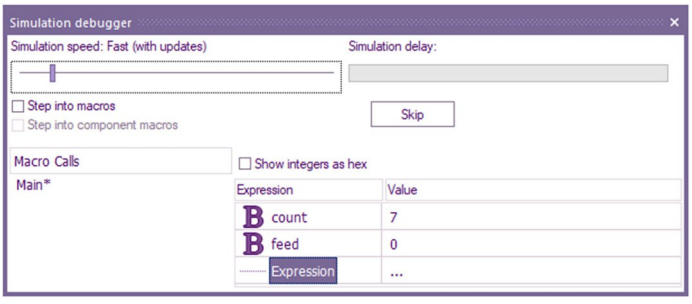
Simulación

Al simular un programa en Flowcode, un rectángulo rojo alrededor de un icono indica el icono que debe ejecutarse a continuación.



Depurador de simulación

Cuando se simula un diagrama de flujo, los valores actuales de las variables utilizadas en el programa pueden verse en esta ventana. Estos valores se actualizan después de simular un comando, a menos que la simulación se ejecute a toda velocidad ('Rápida (sin actualizaciones)').

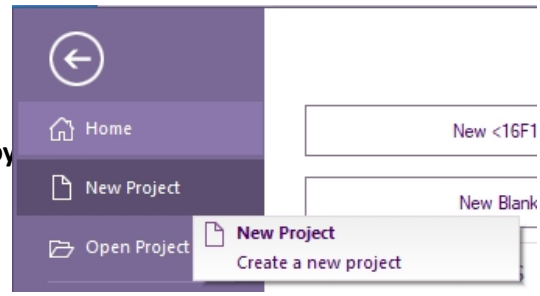


Si simula un diagrama de flujo y luego pulsa el botón de pausa, puede hacer clic en las variables de esta ventana para cambiar su valor. Esto le permite probar su diagrama de flujo en condiciones conocidas.

La ventana también muestra la macro actual que se está simulando en la sección "Llamadas de macro", útil cuando una macro llama a otra durante el proceso de simulación.

Si no dispone de una licencia profesional, la opción de paso a macros de componentes no estará visible.

Iniciar un nuevo diagrama de flujo



- Cree un nuevo organigrama seleccionando **Archivo > Nuevo proy**
- Seleccione el microcontrolador al que desea apuntar de la lista presentada.
- Haga clic en el botón "Nuevo proyecto integrado".

Abrir un proyecto existente

Hay varias formas de abrir un proyecto Flowcode existente:

- **Seleccione la opción Abrir del menú Archivo (Archivo > Abrir proyecto)**

o

- Seleccione el archivo de la lista de archivos utilizados más

recientemente en el menú Archivo. o bien

- Haga doble clic en un archivo Flowcode (.fcfx) en el Explorador de Windows para iniciar Flowcode y abrir el archivo.

Guardar un organigrama

Para guardar un diagrama de flujo, seleccione las opciones "Guardar" o "Guardar como" del menú Archivo **(Archivo > Guardar / Guardar como).**

Los diagramas de flujo deben guardarse antes de compilarlos en C o transferirlos a un microcontrolador.

Guardar imágenes de diagramas de flujo

Para guardar una imagen del diagrama de flujo activo en ese momento, seleccione "Guardar la macro actual como imagen san" en el submenú "Exportar".

-en el menú "Archivo" **(Archivo > Exportar > Guardar la macro actual como imagen).**

Esta función guarda una imagen del programa en cualquier archivo del formato elegido de la lista:

- Mapa de bits (*.bmp);
- JPEG (*.jpg;*.jpeg);
- GIF (*.gif);
- PNG (*.png).

Tenga en cuenta que la tasa de zoom actual se utiliza para determinar la resolución de la imagen guardada. Si necesita imágenes de alta calidad para imprimir, aumente la tasa de zoom.

Desde el menú "Exportar", también tiene la opción de guardar la imagen actual del "Panel 2D", del "Panel de control 2D" o del "Panel del sistema 3D" **(Archivo > Exportar > Guardar el Panel 2D actual como imagen / Guardar el Panel de control 2D actual como imagen / Guardar el Panel del sistema 3D actual como imagen).**

Estas imágenes pueden guardarse en cualquier formato de archivo elegido de la lista:

- Modelo (*.mesh)
- Mapa de bits (*.bmp)
- JPEG (*.jpg;*.jpeg)
- GIF (*.gif)
- PNG (*.png)Modelo (*.mesh)

Ajustes globales

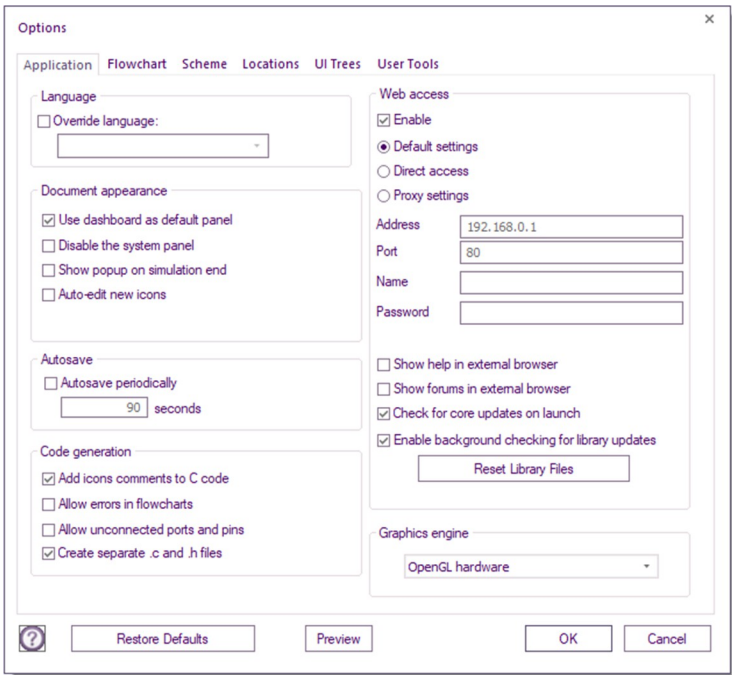
El menú **Archivo** también incluye una **Configuración global** para configurar la aplicación y el diagrama de flujo **(Archivo > Configuración global)** A continuación, seleccione la pestaña correspondiente.

Ficha Aplicación

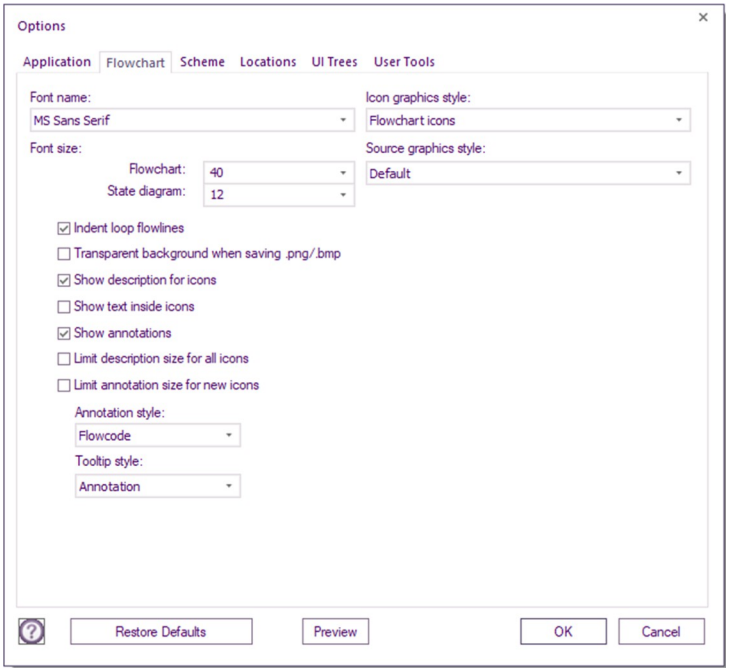
sta pestaña permite configurar los parámetros generales de la aplicación, como el idioma, el aspecto del documento, la función de autoguardado, las opciones de generación de código y el acceso web.

El motor gráfico OpenGL puede configurarse aquí como modo hardware o software.

La opción **Anular idioma** permite al usuario anular la configuración predeterminada del idioma de Flowcode y mostrar Flowcode en un idioma especificado. Para ello, seleccione el idioma de entre los disponibles en la lista desplegable y reinicie Flowcode. Lo hará en el idioma seleccionado, siempre que se haya instalado el paquete de idiomas correspondiente.

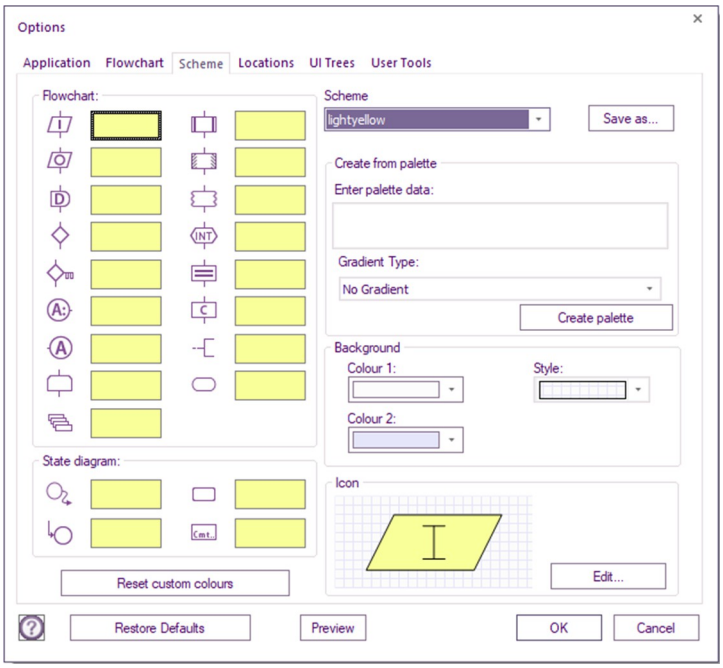


Ficha Diagrama de flujo



Esta pestaña permite configurar los estilos de visualización del organigrama, el tamaño del texto y la fuente.
Personalización del estilo de las anotaciones y de

Ficha Esquema

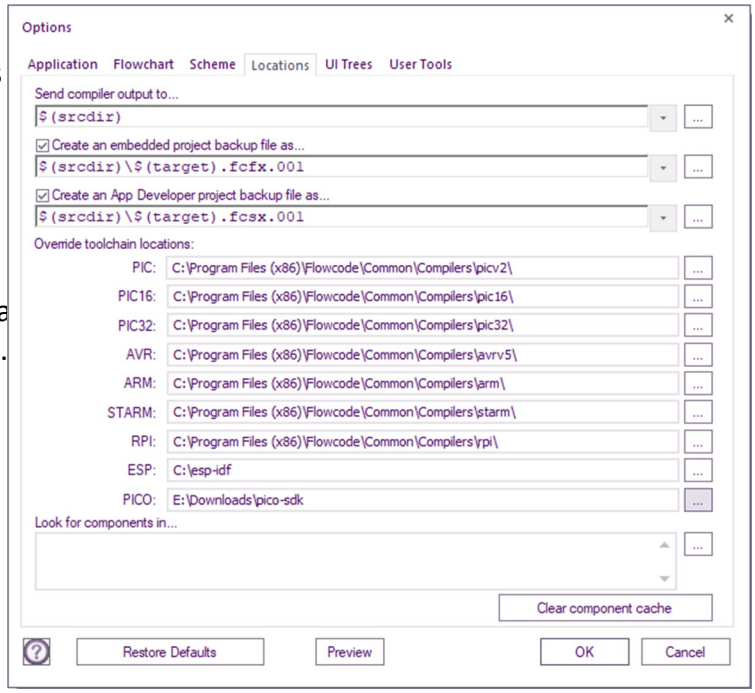


Esta pestaña contiene la configuración para cambiar la apariencia del organigrama, incluidos los colores y gráficos de los iconos, los colores y patrones de fondo, etc.

Ficha Ubicaciones

Esta pestaña permite configurar los nombres de los archivos de copia de seguridad y la ubicación de los directorios de la cadena de herramientas.

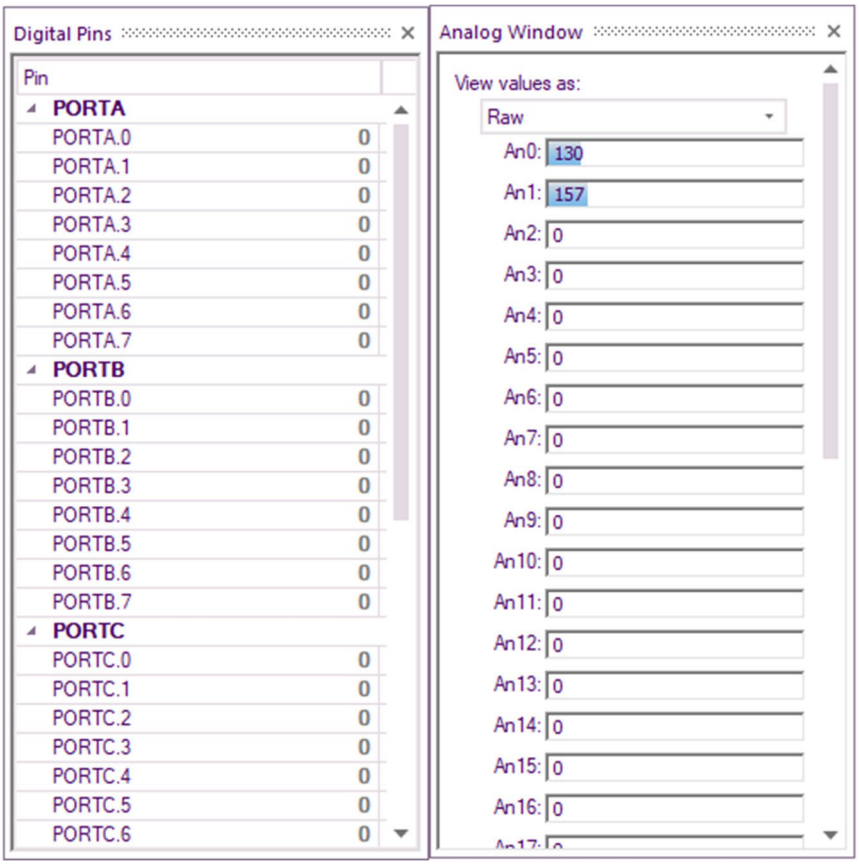
Se pueden añadir directorios adicionales para la ubicación de componentes personalizados.



Ver Ventanas (Simulación)

Entradas analógicas y pines digitales

Los valores de las entradas analógicas se pueden ajustar y los pines digitales se pueden supervisar y ajustar a través de las ventanas habilitadas desde **Ver > Entradas analógicas**
y
Ver > Clavijas digitales



El menú Ver

Esto dicta qué paneles y barras de herramientas aparecen en el espacio de trabajo, una función útil cuando se intenta simplificar su aspecto.
También dispone de un menú **Zoom**, que permite mostrar más iconos en la ventana del espacio de trabajo que cuando se utiliza la configuración de zoom por defecto.

El ajuste actual del zoom se muestra en el submenú Zoom y en la parte derecha de la barra de estado, en la parte inferior de la ventana Flowcode.

El tamaño de cada icono viene dictado por el nivel de zoom: para iconos más grandes, acércalos; para iconos más pequeños, aléjalos. Utilice la función Vista previa de impresión para optimizar el aspecto de su diagrama de flujo en el papel.

También se puede acceder al menú **Zoom** haciendo clic con el botón derecho del ratón en

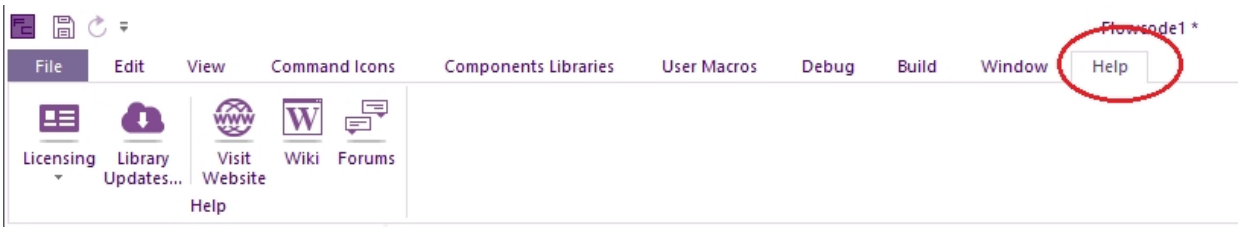
el área de trabajo del organigrama. Atajos de las teclas de función:

- Aumentar Zoom (F3) - aumenta el tamaño del zoom en un 5%;
- Disminuir zoom (F2) - reduce el tamaño del zoom en un 5%;
- Zoom por defecto (F4) - ajuste el zoom al 75%;
- Zoom para ajustar - Amplía para ajustar todo el diagrama de flujo a la ventana actual;
- Zoom para ajustar la anchura - Amplía el diagrama de flujo para ajustarlo a la anchura de la ventana.

Obtener ayuda con Flowcode

Flowcode tiene en su interior y en línea un extenso **wiki** al que se puede acceder a través del menú de la barra de herramientas Ayuda o a través de un navegador de Internet y visitando esta página:

http://www.flowcode.co.uk/wiki/



Además, cada componente de Flowcode tiene una página en la **wiki** en la que se explican todas las macros que contiene y, normalmente, también se incluyen algunos ejemplos.

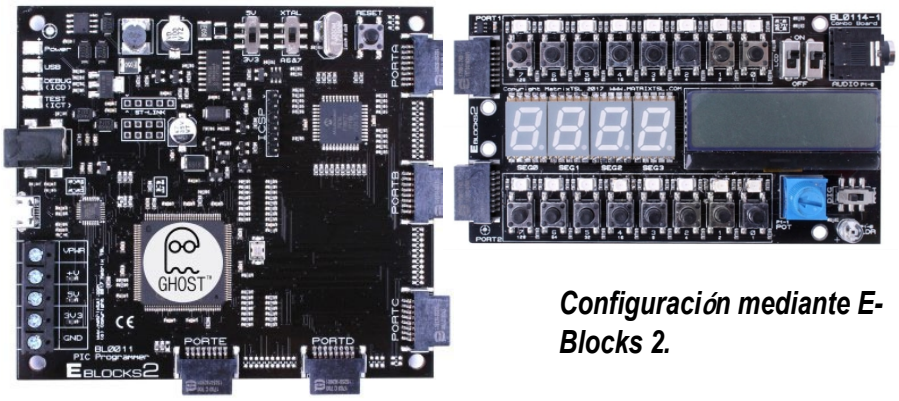
Para acceder a la ayuda de los componentes, basta con hacer clic con el botón derecho del ratón en cualquier componente del panel 2D o 3D y seleccionar **Ayuda**.

Desde aquí se puede ver:

- Explicación del componente
- Algunos ejemplos de uso del componente
- Referencias de macros que explican qué es cada macro más los parámetros y valores de retorno.

Sección 4: Flowcode

Primer programa



Configuración mediante E-Blocks 2.

Añadir salidas digitales - Encender el LED

Crea un programa que encienda un **LED** conectado al microcontrolador. Este programa introduce el tema de cómo controlar una salida digital.

El tutorial proporciona un enfoque claro, paso a paso que le permite crear su primer programa utilizando Flowcode. Puede ejecutarse en el modo de simulación de Flowcode antes de compilarlo en la placa para pruebas y desarrollo.

Nota: Este tutorial se refiere a la configuración de puertos (puertos A y B)

como se utiliza con PIC. Para los usuarios de Arduino, por favor, utilice los

puertos C y D según corresponda.
(Puerto C en el Arduino 'Maps'to Puerto A de la placa Combo).

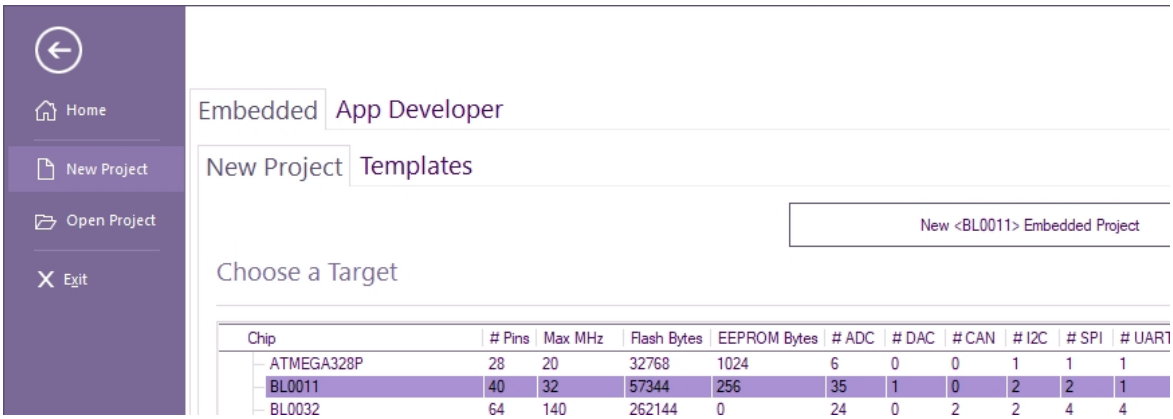
Iniciar un nuevo proyecto

Seleccione "Nuevo proyecto" en la pantalla de bienvenida o a través del menú **(Archivo > Nuevo proyecto).** En la pestaña "Embedded", elija un dispositivo de destino o una placa de desarrollo.

Observará que la lista de selección incluye detalles sobre las características y periféricos de cada objetivo. Esto resulta útil a la hora de seleccionar un dispositivo para un proyecto concreto.

Para nuestro nuevo proyecto, si estamos utilizando por ejemplo la placa de desarrollo PIC MatrixTSL E-blocks2, elige el objetivo BL0011 de la lista "Objetivos libres".

Para los usuarios de Arduino, seleccione una placa de desarrollo Arduino adecuada.

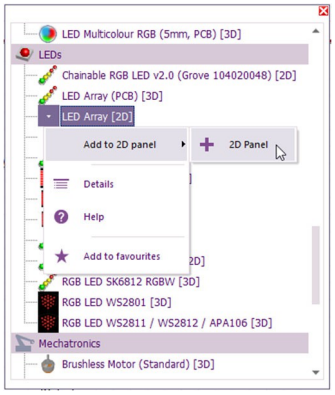


En este caso, la elección de destino también selecciona el dispositivo 16F18877 correcto y preestablece los valores correctos para la frecuencia del oscilador del reloj y

otros ajustes. Haga clic en el botón "Nuevo <BL0011> Proyecto integrado" para

iniciar el proyecto.

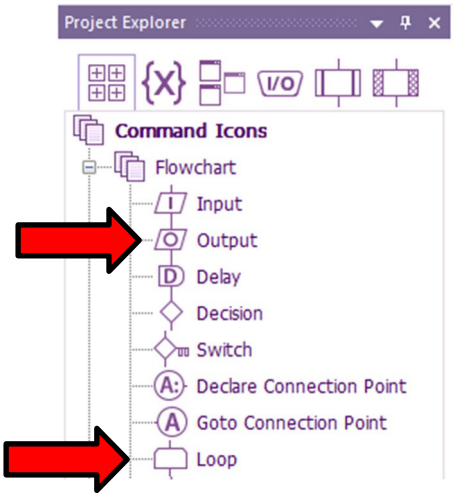
CONSEJO: El dispositivo de destino del proyecto puede cambiarse posteriormente a través del menú **(Build > Project Options)**

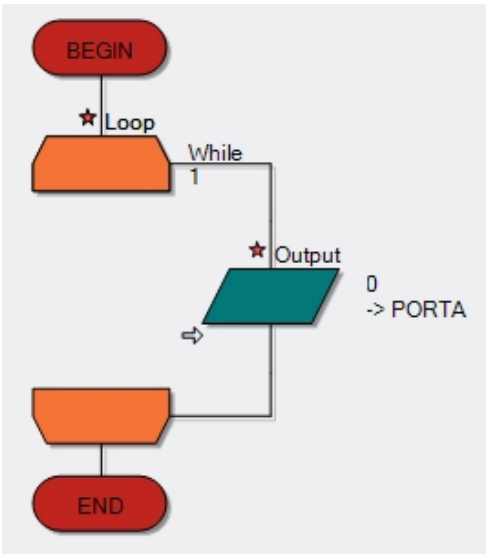


Añade una matriz de LEDs [2D] al panel 2D.

La matriz de LEDs se encuentra en **Salidas** en la Barra de Herramientas de Bibliotecas de Componentes.

(Bibliotecas de componentes > Salidas > Matriz de LED [2D]> Añadir a panel 2D)





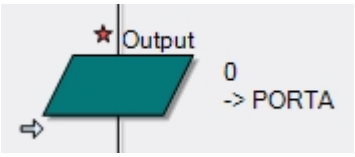
Cree un organigrama.

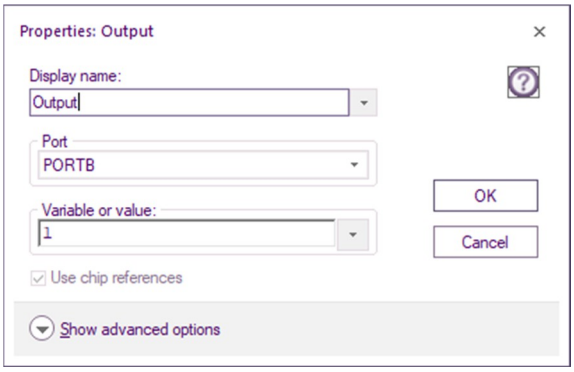
Mueva el cursor sobre el icono **Bucle**, en la barra de herramientas Icono. Haga clic y arrástrelo hasta el área de trabajo. Mientras lo arrastra, el cursor normal se transforma en un pequeño icono. Muévalo entre los iconos 'BEGIN' y 'END'. Al hacerlo, aparecerá una flecha que le indicará dónde se colocará el icono **de bucle**.
Suelte el botón del ratón para colocar el icono entre las casillas "BEGIN" y "END".

Añada un icono de **Salida** dentro del bucle en el diagrama de flujo de la misma manera.

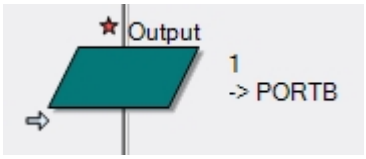
CONSEJO: *Los colores de los iconos de su sistema pueden ser diferentes.*

Cambiar la configuración de los puertos Haga doble clic en el icono de salida que ha colocado en su organigrama y aparecerá el cuadro **Propiedades**.





Seleccione Puerto B. Introduzca un valor de 1.



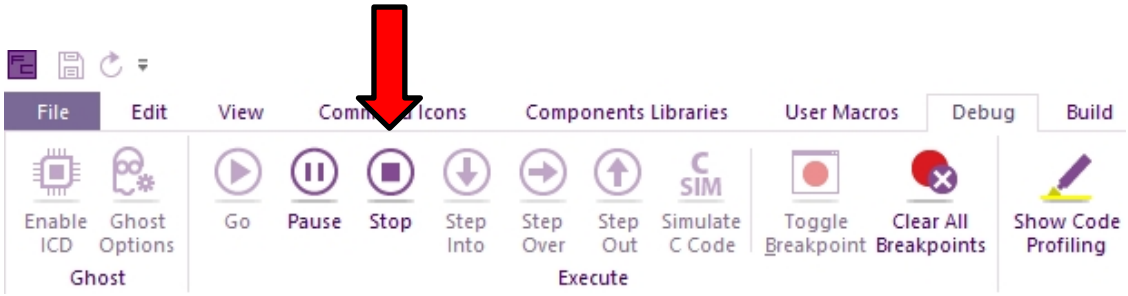
(Has hecho esto porque los LEDs de tu panel del sistema 3D están actualmente conectados al puerto B, así que estamos enviando la **Salida** al mismo puerto).

Ejecuta la simulación.
Selecciona el icono **Ir** de la barra de menú **Depurar** y la simulación del LED se iluminará en el panel 3D del sistema.

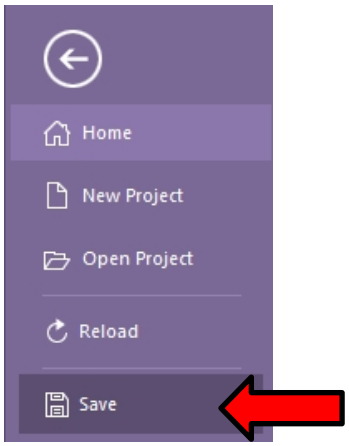
Ir (F5)

Modo simulación.

Detener (Mayús+F5)



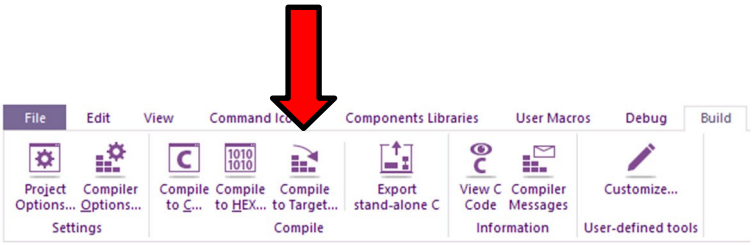
CONSEJO: *Recuerde detener su simulación antes de hacer cualquier otra cosa. (Si Flowcode no está haciendo lo que usted espera, compruebe que no ha dejado accidentalmente su simulación en ejecución).*



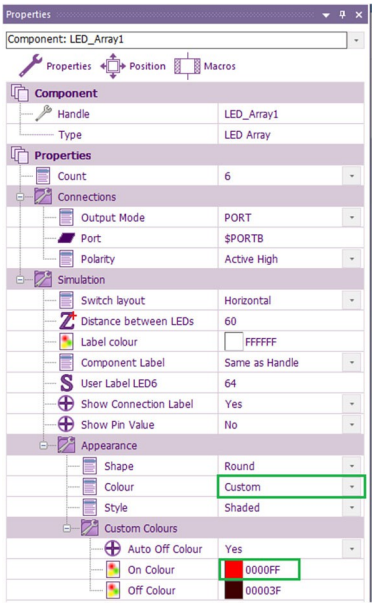
Guarde su programa **(Archivo > Guardar)**

Conecta tu placa de desarrollo objetivo a una fuente de alimentación.
Conecta el cable de programación USB a tu PC.

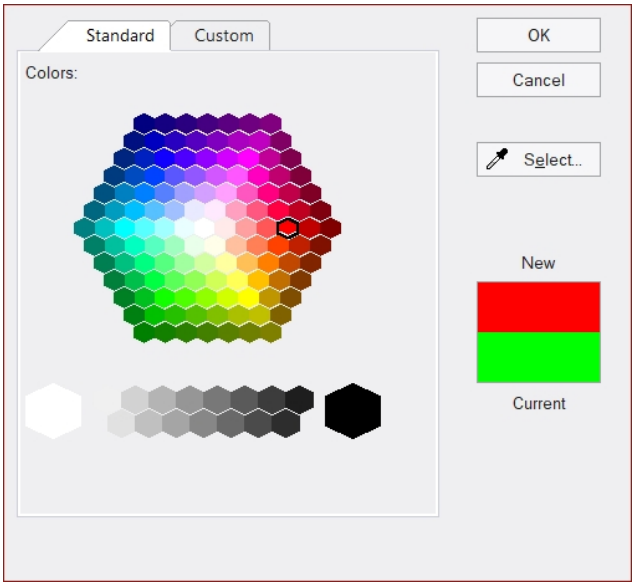
Haga clic en el botón **Compilar en destino** de la ventana **Compilar** como se muestra: **(Compilar > Compilar en destino)**



Cambios para probar después de encender con éxito su LED.
Resalta la imagen de la matriz de LEDs en el panel del sistema 3D y haz click derecho para seleccionar las **Propiedades**. Aquí puedes cambiar el número de LEDs en tu matriz cambiando el valor bajo **count**. Intenta cambiar el color de los LEDs en la simulación como se muestra a continuación.



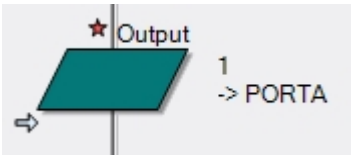
SeFngs de propiedad para 6 LED rojos.

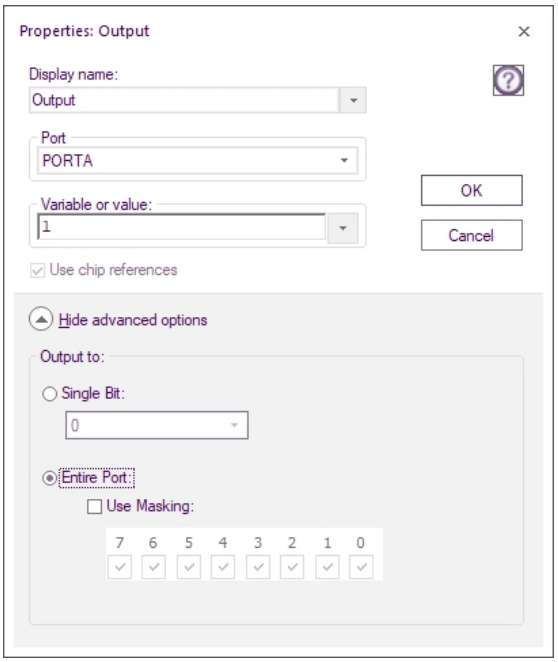


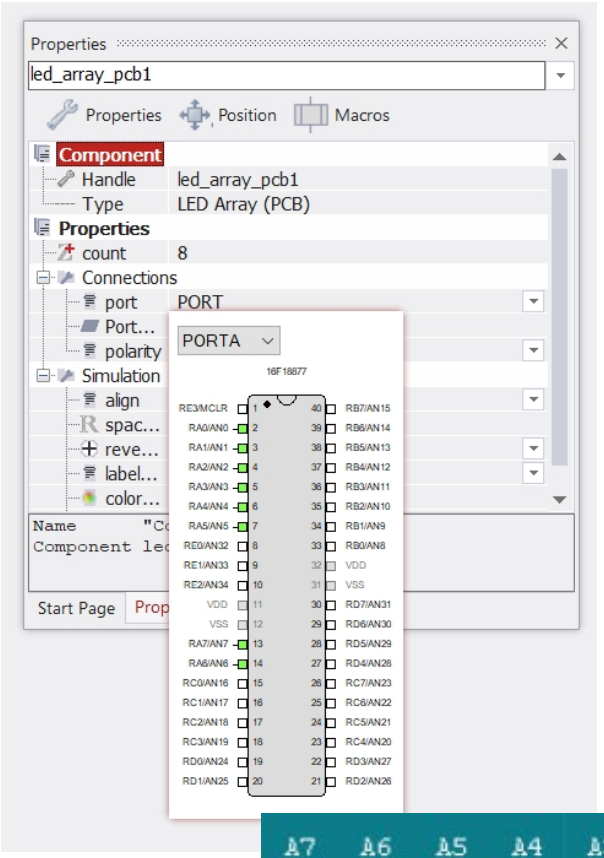
6 LED rojos en simulación.

Cambiar la configuración del puerto.

Abre las propiedades del icono de salida (doble clic) y cambia la configuración del puerto a **Puerto A.** Resalte la imagen de la matriz de LED en el panel del sistema 3D y haga clic con el botón derecho para seleccionar las **Propiedades**, y cambie la configuración de Puerto a **Puerto A**.

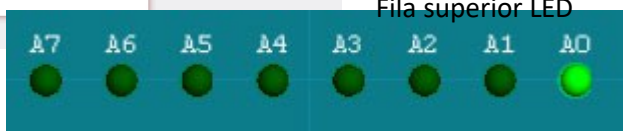


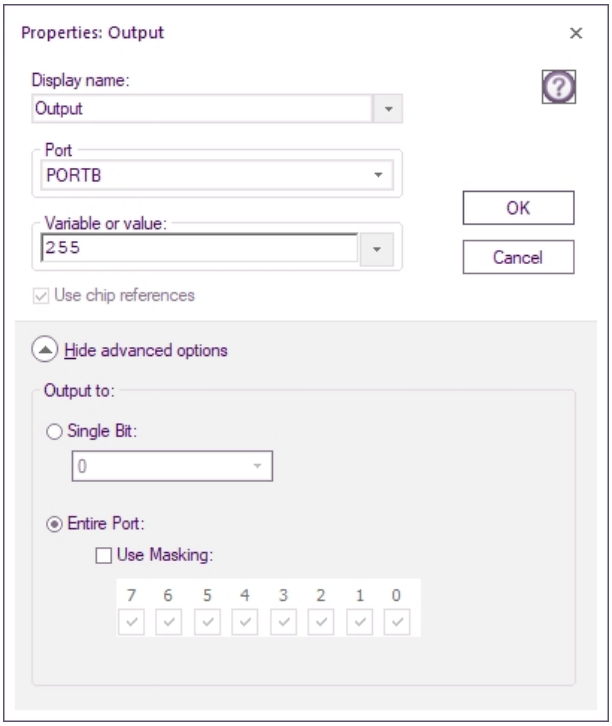


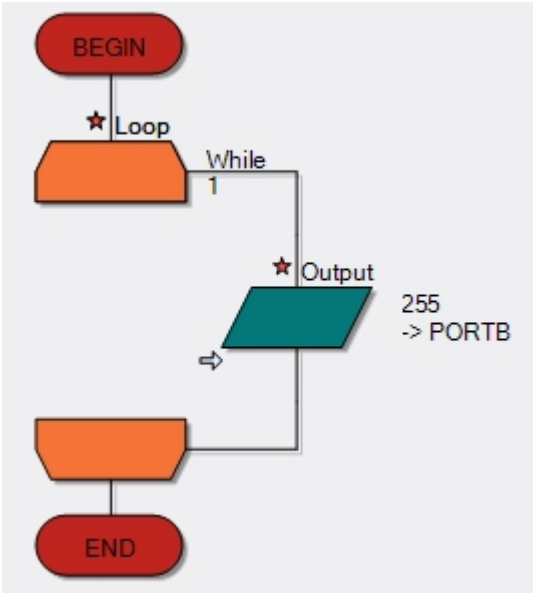


Ejecutar en modo de simulación y luego compilar a chip. Deberías ver que se enciende el primer LED de la otra fila.

Fila superior LED







Puedes practicar cambiando los puertos cambiándolos de nuevo al **puerto B**. Cambia el **valor** de 1 a **255**. Prueba en modo simulación y luego compila al chip (los 8 LEDs se encienden). Experimenta con otros valores. (***SUGERENCIA: Consulte la hoja de trabajo Sistemas numéricos).***

Números binarios

Los dispositivos electrónicos digitales no pueden manejar números decimales (0, 1, 2, ..9 etc.). En su lugar, utilizan el sistema binario, que utiliza sólo dos números 0 y 1. El número 1 podría representarse mediante una señal de alta tensión, mientras que el número 0 podría ser de baja tensión.

Decimal	Igual en binario
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010

La tabla de al lado muestra cómo se comparan los dos sistemas numéricos:

El sistema decimal utiliza diez números: 0, 1, 2, 3, 4, 5, 6, 7, 8 y 9. Al llegar al último, el "9", empezamos de nuevo con el "0", pero añadimos otro número delante. Al llegar al último de ellos, el "9", empezamos de nuevo con el "0", pero añadimos otro número delante. Por ejemplo, después del "8" y el "9" viene el "10", y después del "18" y el "19" viene el "20", y así sucesivamente. Cuando lleguemos al 99, ambos volverán a ser 0, pero con un 1 delante, para formar el 100.

BINARY VALUE				
16	8	4	2	1

Decimal	Igual en binario
1	1
2	10
4	100
8	1000

En binario ocurre lo mismo, pero mucho más a menudo, porque sólo se utilizan "0" y "1". La cuenta ascendente empieza con un "0", luego un "1", y de nuevo un "0" con un "1" delante, lo que da "10" (no diez, ¡son dos!). A continuación viene "11" (tres) y se empieza de nuevo con dos "0" pero con un "1" delante, para dar "100" (cuatro) y así sucesivamente.

Observa que cada vez que el "1" binario se desplaza un lugar a la izquierda, se duplica el valor del número en decimal, como muestra la segunda tabla. Podemos utilizar esta idea para convertir entre sistemas numéricos.

CONSEJO: En cualquier número binario, el bit situado en el extremo izquierdo, el bit más significativo (MSB), tiene el valor más alto. El que está en el extremo derecho, el bit menos significativo (LSB), tiene el valor más bajo.~

Números hexadecimales

El hexadecimal, abreviado "hex", es una forma más cómoda que el binario (para los humanos) de representar números.

- Un dígito binario es 0 o 1.
- Un dígito decimal varía entre 0 y 10.
- Un dígito hexadecimal tiene dieciséis estados posibles.

Está claro que dieciséis estados es un problema, ya que sólo tenemos los dígitos del 0 al 9. Para evitarlo, utilizamos las letras de la A a la F para proporcionar los seis dígitos adicionales necesarios. Para evitarlo, utilizamos las letras de la A a la F para obtener los seis dígitos adicionales necesarios.

Trabajar con el número binario de ocho dígitos es una convención práctica, ya que los ordenadores (y la MCU PIC) almacenan la información en grupos de ocho bits.

Una sola celda de memoria dentro de un dispositivo PIC puede almacenar un número entre 0000 0000 y 1111 1111. En decimal este rango es de 0 a 255. El equivalente en hexadecimal es de 0 a FF.

CONSEJO: Puede introducir un número hexadecimal en Flowcode precediéndolo de '0x'en cualquiera de los cuadros de diálogo.

Constructos de codificación - Sistemas numéricos

Binary value					Decimal value
16	8	4	2	1	
				1	1
			1	0	2
		1	0	0	4
	1	0	0	0	8
so					
		1	1	1	7
	1	0	0	1	9
1	0	1	0	0	20
1	1	1	1	1	31

Una sola celda de memoria dentro de un dispositivo PIC puede almacenar un número entre 0000 0000 y 1111 1111. En decimal este rango es de 0 a 255. El equivalente en hexadecimal es de 0 a FF.

Decimal	Binary	Hex
0	00000000	0
1	00000001	1
2	00000010	2
3	00000011	3
4	00000100	4
5	00000101	5
6	00000110	6
7	00000111	7
8	00001000	8
9	00001001	9
10	00001010	A
11	00001011	B
12	00001100	C
13	00001101	D
14	00001110	E
15	00001111	F

Tareas

Completa la siguiente tabla

a. Sombreado de los LED que iluminan, para las tres primeras filas.

b. Averiguar qué número produce los patrones de LED que se muestran en las tres últimas filas.

Number sent to output	LED array
51	
204	
195	
_____	
_____	
_____	

Utiliza Flowcode para:

a. Comprueba tu trabajo a partir de la tabla anterior utilizando Flowcode.

b. Introduzca un número hexadecimal en Flowcode precediéndolo de '0x'en cualquiera de los cuadros de diálogo. ¿Puedes iluminar los mismos patrones LED utilizando Hex?

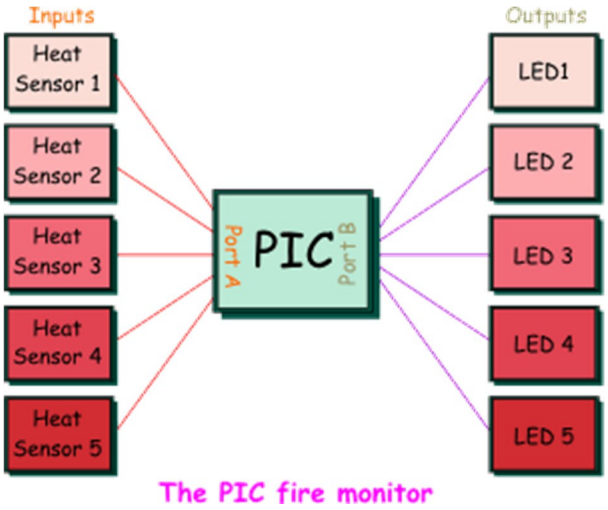
Sección 5:

Flowcode Ejemplos

Todos estos ejemplos pueden probarse utilizando un microcontrolador PIC o Arduino.

Los usuarios de Arduino deben familiarizarse con **los ajustes de Arduino** en el Apéndice 1, y ajustar cualquier configuración de puerto en consecuencia.

Ejemplo 1: Añadir entradas digitales - ¿Dónde está el fuego?



¡El escenario!

Un edificio grande tiene varios sensores térmicos en su sistema de alarma contra incendios. Cuando se produce un incendio, los bomberos necesitan saber dónde está el fuego. En otras palabras, necesitan saber qué sensor térmico ha disparado la alarma.

El sistema está controlado por un dispositivo PIC. Hay cinco sensores de calor, conectados como entradas al puerto A. El puerto B está configurado como puerto de salida y conectado a un conjunto de cinco LED. Si un sensor de calor detecta un incendio, se enciende el LED correspondiente.

Configuración del organigrama

Para los usuarios de Arduino, utilice los puertos C y D según corresponda.
(Puerto C en el Arduino 'Maps'to Puerto A de la placa Combo).

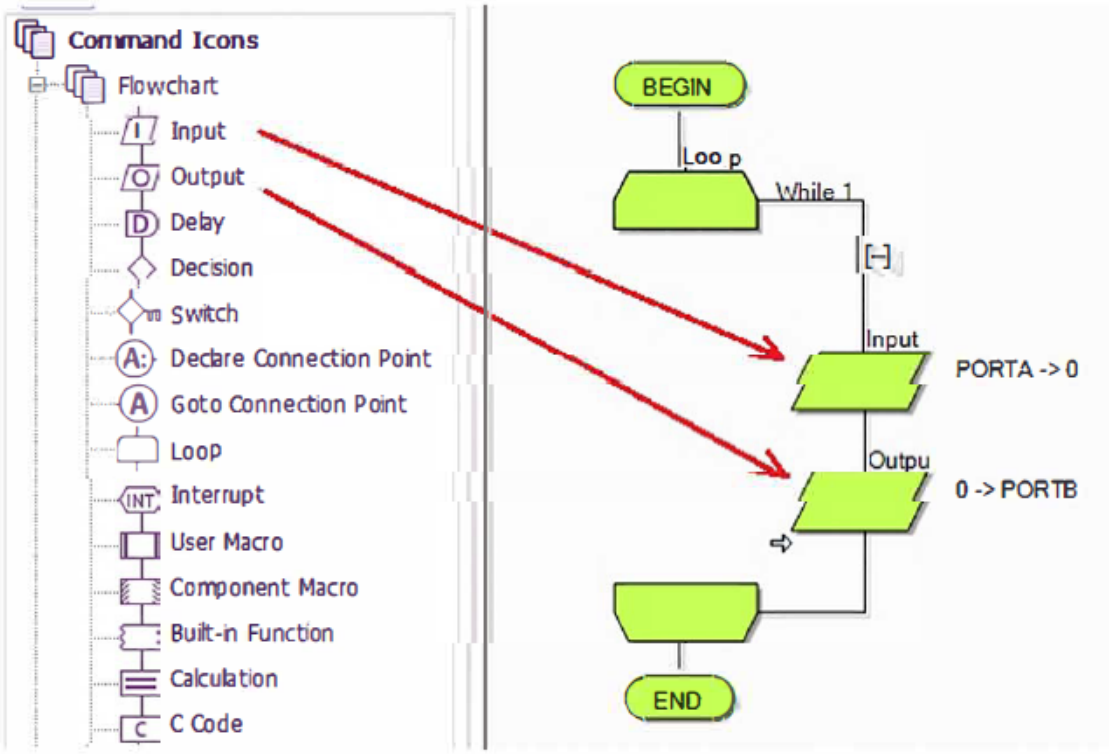
Abra Flowcode y cree un nuevo proyecto adecuado para la placa que está utilizando.

Arrastre el **icono Bucle**, el **icono Entrada** y el **icono Salida**
en su organigrama desde

la barra de herramientas de iconos para crear un

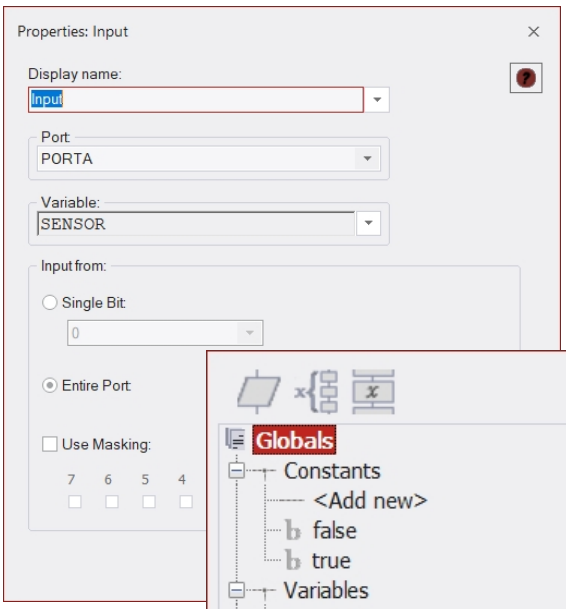
Diagrama de Flujo como se muestra. Establezca la

Entrada en el puerto A y **la Salida** en el puerto B.



Creación de las variables

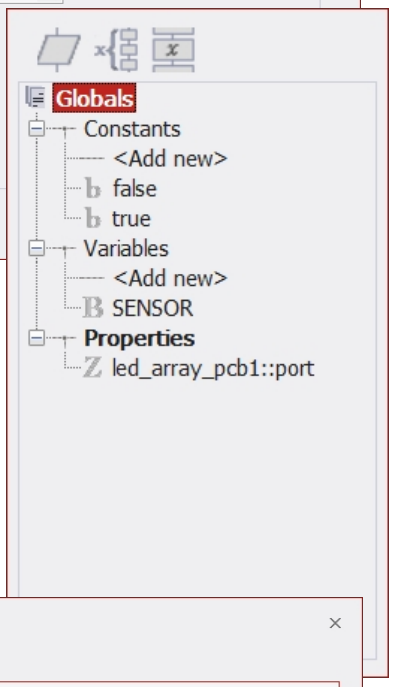
Haga clic con el botón derecho del ratón en el icono de entrada y seleccione "Propiedades" en el menú. Aparecerá el cuadro de diálogo Propiedades de entrada, que se muestra al lado. Esto nos permite añadir una "variable". Pero, ¿qué es una variable?



Una variable es un lugar donde podemos almacenar información, en particular, información que cambia a medida que nuestro programa se ejecuta.

En este caso, es el número del sensor de calor el que dispara la alarma. Puede ser el sensor 1 el que se dispare, o el sensor 5.... .

Vamos a utilizar una variable llamada **SENSOR** para almacenar la información sobre qué sensor se ha disparado.

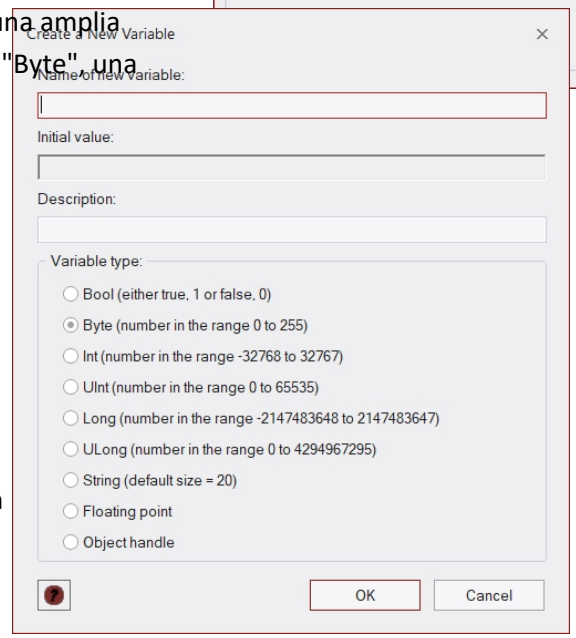


Haga clic en la flecha situada junto a la casilla "Variable:".

Aparecerá el siguiente cuadro de diálogo:

Ahora pasa el ratón por encima de la palabra 'Variables' y aparecerá la flecha. Haz clic en ella y selecciona "Añadir nueva".

Aparece otro cuadro de diálogo, que se muestra al lado, que ofrece una amplia selección de tipos de variables. Por ahora, acepte el tipo por defecto "Byte", una variable que puede almacenar números de "0" a "255".



Escriba el nombre "SENSOR" (sin comillas) como nombre de la nueva variable y pulse el botón 'Aceptar'. Ahora aparece en la lista de variables que puede utilizar el diagrama de flujo.

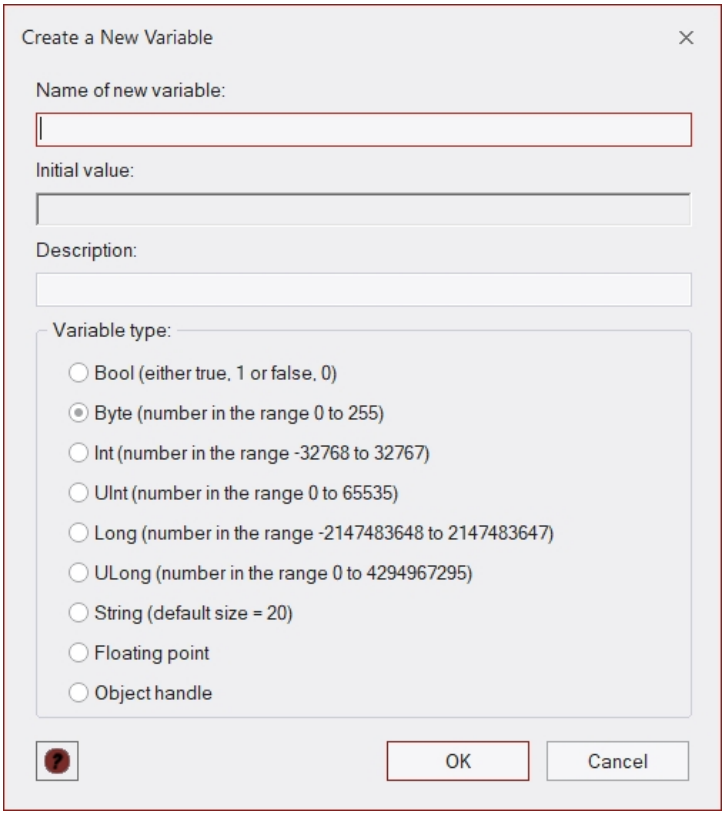
Haga doble clic en el nombre de la variable para utilizarla o, alternativamente, haga clic y arrastre el nombre al cuadro de variables.

Ahora verá de nuevo el cuadro de 'Propiedades' de entrada. Fíjate en que tienes que decirle al sistema qué puerto vas a utilizar para introducir los datos que necesita el sistema. Por el momento está configurado en el puerto A, y vamos a dejarlo así.

En este caso, el sistema necesita monitorizar los sensores de calor, por lo que cada sensor se conectará a un bit diferente del puerto A. Haga clic en "Aceptar" para cerrar el cuadro Propiedades de entrada.

Más sobre variables

En la sección anterior has añadido una variable al programa utilizando el cuadro de diálogo de variables:



Las señales informáticas consisten en secuencias de "0" y "1" binarios en cada cable. Un grupo de ocho hilos puede transportar ocho "bits" (dígitos binarios) simultáneamente. Esta agrupación de ocho bits, conocida como "byte", se utiliza en gran parte del cableado interno de los microcontroladores y en los registros que almacenan y procesan los datos.

También se utiliza en los subsistemas de memoria. El contenido de un registro de memoria de ocho bits puede variar de "0" a "255".

Una variable dentro de Flowcodec puede configurarse para utilizar sólo un registro de memoria o más de uno.

Variables Flowcode:

Flowcode ofrece ocho tipos diferentes de variables:

- una variable 'Bool'(Booleana) puede ser '1'o '0'(verdadero o falso);
- un único registro, conocido como variable "Byte", puede almacenar números de "0" a "255";
- un registro doble, conocido como variable 'Int'puede almacenar números de '-32768'a '+32767';
- un registro doble también puede ser sin signo, cuando se conoce como variable 'UInt'que puede almacenar números de '0'a '65535';
- un registro cuádruple, conocido como variable 'Long', puede almacenar números desde '-2147483648'hasta '2147483647';
- un registro cuádruple también puede ser sin signo, cuando se conoce como variable 'ULong', que puede almacenar números de '0'a '4294967295'.

CONSEJO: Utilice una variable 'Byte'para contadores sencillos y para variables que no superen el valor '255'. Es la más económica en términos de espacio de memoria y también la más rápida. Los procesos matemáticos que implican dos bytes (a menudo denominados 'aritmética de 16 bits') tardan más en ejecutarse. Un registro múltiple, conocido como variable 'Cadena', puede consistir en un número de variables 'Byte' - el valor por defecto en Flowcode es 20.

Otras cuestiones variables:

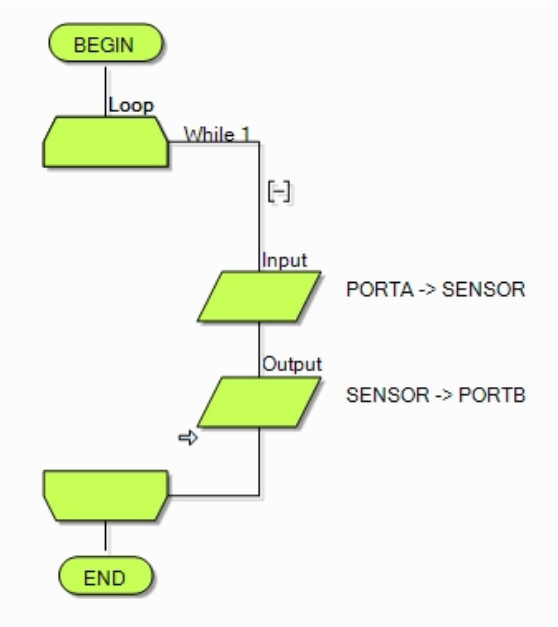
También se pueden utilizar números de coma flotante (que contienen un punto decimal en algún lugar), aunque representan una gama de valores mucho más amplia que los enteros. Sufren una pérdida de precisión en rangos grandes.

Por último, un "manejador de objeto" se utiliza para hacer referencia a un dato más complicado (como un archivo, un componente o un bloque de texto) cuyo formato interno se desconoce.

¿Por qué preocuparse?

El número de registros dentro de un microcontrolador es limitado, y en aplicaciones más grandes el número y los tipos de variables deben gestionarse cuidadosamente para garantizar que haya suficientes. Al descargar un programa, las variables en Flowcode se implementan en la parte de Memoria de Acceso Aleatorio (RAM) del dispositivo PIC. En el 16F18877 hay 4096 Bytes de memoria. Esto significa que puede tener 4096 variables 'Byte', 2048 variables 'Int' o 204 'Strings' cada una con veinte 'Bytes' o caracteres.

Configuración de las salidas



- A continuación, haga clic con el botón derecho del ratón en el icono de Salida y seleccione 'Propiedades'o simplemente haga doble clic sobre él. Aparecerá el cuadro Propiedades de salida.
- Haga clic en la flecha, junto a la casilla 'Variable:'. Verás que aparece la variable 'SENSOR'.
- Haga doble clic sobre la palabra 'SENSOR'o haga clic y arrástrela hasta la casilla 'Variable:'.
- El cuadro de Propiedades de Salida muestra ahora que el sistema está configurado para dar salida a cualquier dato almacenado en la variable 'SENSOR'. Cambia el puerto utilizado al puerto B, haciendo clic en la flecha, ,
- en la ventana de puertos y, a continuación, haga clic en 'PORTB'en el menú que se abre.
- Haga clic en "Aceptar" para cerrar el cuadro Propiedades de salida
- El diagrama de flujo debería tener ahora este aspecto:
- Fíjese en las flechas de los iconos. Muestran que la información fluirá desde el puerto A hacia el diagrama de flujo, a través de "SENSOR", (icono de entrada) y desde el diagrama de flujo, a través de "SENSOR", hacia el puerto B (icono de salida).

Añadir los LED

- Ahora haz clic en el botón Salidas y selecciona el icono Matriz de LEDs. Haz clic y arrástralo al Panel de Sistema.
- Cambie la propiedad "Count" de la sección "Simulation" al valor "5" haciendo clic en la casilla situada junto a la propiedad "Count" y utilizando el teclado para introducir el valor.
- Haz clic junto a "Puerto" en la sección "Conexiones" para abrir una vista interactiva del chip, que muestra los pines compatibles.
- Haz clic en el menú desplegable y selecciona la opción 'PUERTO B'. Ahora has conectado los LEDs a los pines del puerto B.

(Para los usuarios de Arduino, utilice los puertos C y D según corresponda).

Añadir los interruptores

- Vas a utilizar cinco interruptores para simular los cinco sensores de calor. El interruptor que está "encendido" (cerrado) es el sensor de calor que ha disparado la alarma de incendios.
- Haga clic en el botón "Entradas" y seleccione la matriz de interruptores. Arrástralo a un lugar adecuado del Panel de Sistema.
- Haga clic en la casilla situada junto a la propiedad "Count" y cambie el valor a "5". Compruebe que el componente está conectado a 'PORTA'.

Simulación del programa

- Haz clic una vez en el botón 'Step Into'. Aparecerá la ventana 'Simulation Debugger', pero ignórala por ahora.
- Mueva el cursor sobre uno de los interruptores y haga clic, para simular la detección de un incendio. El gráfico del interruptor cambia a la posición de cerrado. Haga clic en el botón 'Step Into'unas cuantas veces más para simular el programa completo.

El programa está terminado y funcionando. Acaba de detectar un incendio que ha activado un sensor de calor. La matriz de LED le indica a usted, o a los bomberos, qué sensor ha detectado el incendio.

Ejemplo 2 - Utilización de bucles

Contar ovejas, mal al principio, ¡pero sin dormirse!

El plan es sencillo: cuando una oveja atraviesa la puerta, rompe un haz de luz. Esto envía un impulso a un sistema de recuento, que entonces suma uno al total almacenado en el sistema.

Mostramos este total en la matriz de LED.

El plan parece sencillo, pero habrá problemas.

(Tenga en cuenta que Flowcode tiene un componente 'Beam Breaker', basado en el 'Collision Detector'. Aunque esto haría un trabajo mucho mejor, por ahora detectamos la interrupción del haz de luz utilizando métodos más básicos).

Configuración del organigrama

Inicie Flowcode e inicie un nuevo diagrama

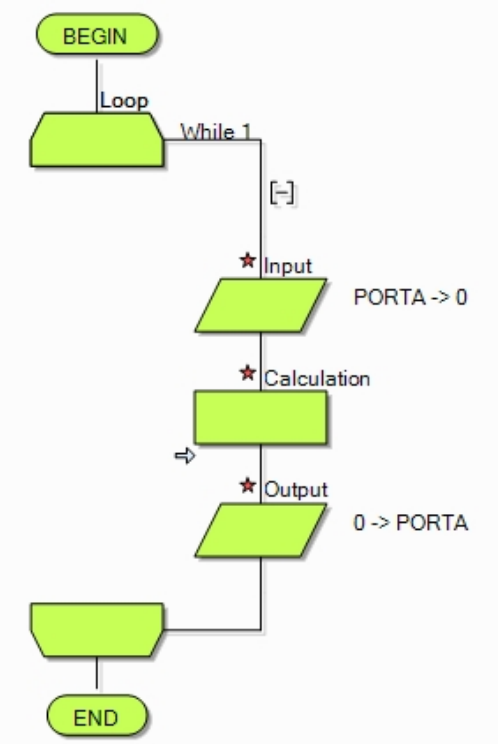
de flujo. Cree el diagrama de flujo que se

muestra al lado.

Contiene un icono de "Bucle" y otro de "Cálculo" que no ha utilizado antes.

Contiene un icono de Entrada y otro de Salida.

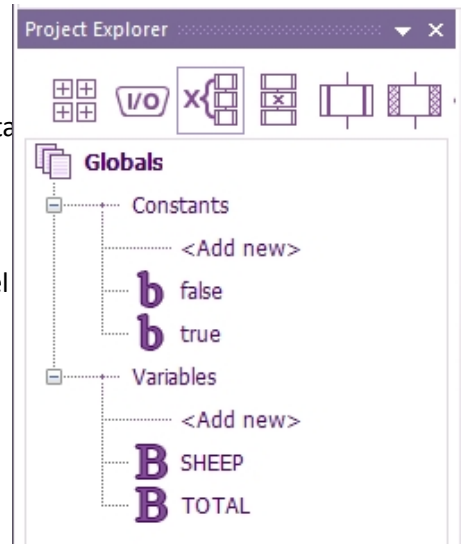
(Para los usuarios de Arduino, utilice los puertos C y D según corresponda



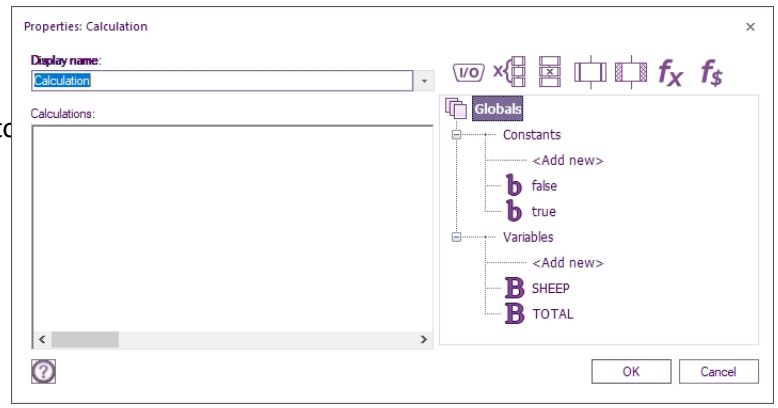
Creación de las variables

Vamos a crear dos variables, una llamada 'OVEJA'y otra llamada 'TOTAL'.

- La variable "OVEJA" mostrará si hay o no una oveja presente.
- La variable 'TOTAL'almacenará el número total de ovejas registradas hasta
- Haga clic en "Ver" en la barra de menús y asegúrese de que la opción "Explorador de proyectos" está seleccionada (Ver > Explorador de proyectos).
- Haga clic en el botón "Globales" situado en la parte superior del panel del
- Pasa el ratón por encima de "Variable:" en el panel del explorador de proyectos y haz clic en "Añadir nueva". Aparecerá el cuadro de diálogo "Crear una nueva variable". Escribe el nombre "OVEJA" y haz clic en 'Aceptar'. Puedes dejar el tipo de variable como "Byte" ya que no habrá muchas ovejas.
- Cree una variable llamada "TOTAL" de la misma manera.



Configuración del cálculo

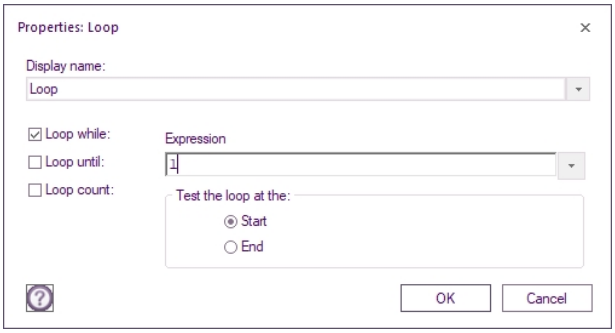


- Haga doble clic en el icono "Cálculo" para abrir el cuadro de diálogo "Propiedades".
- Cambia el "Nombre a mostrar" por "Nuevo to
- Cree el cálculo escribiendo lo siguiente en la ventana 'Cálculos':

  TOTAL = TOTAL + OVEJAS

- Simularemos la ruptura del haz de luz utilizando un interruptor pulsador marcado con 'SW0'en el puerto A bit 0.
- Las propiedades 'Input' están configuradas para almacenar cualquier número que aparezca en el puerto A en el variable llamada 'OVEJA'. Inicialmente, ese número es '0'. Cuando se pulsa el interruptor, el número en el puerto A y almacenado en la variable "SHEEP" es "1". (Con sólo un interruptor, el mayor número que podemos crear en el puerto A es 1).
- Cuando se ejecuta el icono 'Cálculo', el número almacenado en la variable 'OVEJA' se añade a la variable 'TOTAL'. Por lo tanto, cuando una oveja rompe el haz de luz, "TOTAL" se incrementa en 1. Sin ovejas presentes, "TOTAL" permanece sin cambios.
- Pulse el botón "Aceptar" para cerrar el cuadro de diálogo.

Configuración de las propiedades del bucle



- Haga doble clic en el icono "Bucle" para abrir el cuadro de diálogo "Propiedades".

Esto muestra las opciones para controlar el bucle. Junto a la sentencia 'Bucle while:'está la caja de texto de control del bucle, donde se escribe la condición del bucle - el programa continúa el bucle hasta que se cumpla esta condición.

Ejemplos de condiciones de bucle:

- count = 10 (El bucle se ejecuta mientras la variable 'count'= 10)
- count > 4 (El bucle se ejecuta mientras 'count'sea mayor que 4)
- count = preset (El bucle se ejecuta mientras 'count' sea igual a la variable 'preset')

En todos ellos, el bucle continúa mientras la condición del cuadro de texto "Bucle mientras" sea "verdadero".

En programación, "verdadero" tiene un significado especial. Se le asigna un valor numérico de "1" para que una prueba pueda determinar si algo es "verdadero". Del mismo modo, a "falso" se le asigna el valor numérico "0".

La condición por defecto en la caja de texto 'Loop while:'es '1'- esta condición es siempre 'true'y por lo tanto con este valor, el bucle se ejecutará para siempre. Los programas normalmente contienen una estructura de 'bucle para siempre'. Si no lo hacen, el programa terminará de repente y el ordenador se quedará sin hacer nada.

¿Cuándo hacer las pruebas?

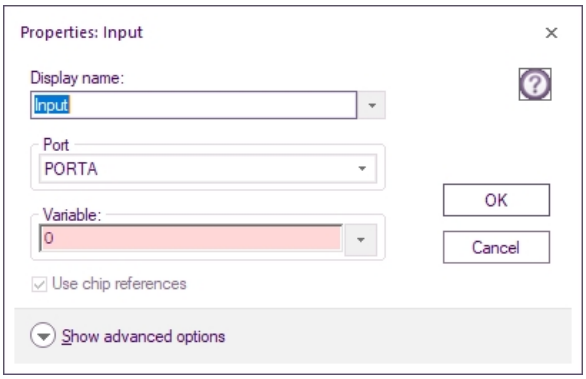
Puede configurar las propiedades para comprobar la condición del bucle al principio del bucle o al final. Entender esta opción es importante. Puede afectar el número de veces que el programa hará un bucle.

Bucle para un número determinado de veces

A veces, sólo quieres ejecutar un bucle durante un número determinado de iteraciones. Para ello, marque la casilla 'Recuento de bucles:'e introduzca el número de bucles que desee en el cuadro de texto asociado.

Configuración de la entrada

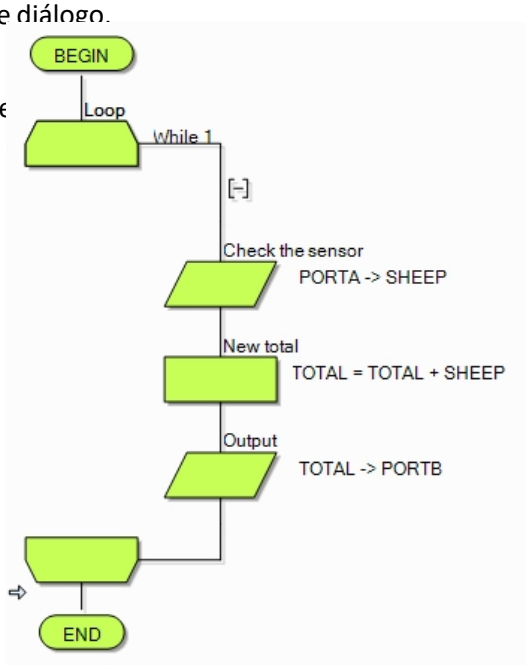
- Haga clic con el botón derecho del ratón en el icono "Entrada" y seleccione "Propiedades" en el menú, para ver el siguiente cuadro de diálogo:
- Cambie el nombre de visualización. Haga doble clic en "Entrada" en el cuadro "Nombre de visualización:" y escriba "Comprobar el sensor".
- Haga clic en juntoa la casilla "Variable:" para abrir el "Gestor de variables".
- Haga doble clic en la palabra "OVEJA" para insertarla en la casilla "Variable:".
- Por defecto, la entrada es el puerto A, que es el que queremos. Haz clic en "Aceptar" para cerrar el cuadro de diálogo.



Configurar la salida

- Haga doble clic en el icono "Salida" para abrir el cuadro de diálogo "Propiedades" de la salida.
- Haga clic en la casilla "Variable:".
- Haga doble clic en la palabra "TOTAL" para insertarla en la casilla "Variable:".
- En la casilla'Propiedades'de la salida, cambia el puerto utilizado a 'PORTB'.
- Haga clic en "Aceptar" para cerrar el cuadro de diálogo.

El diagrama de flujo debería tener ahora este aspe



(Para los usuarios de Arduino, utilice los puertos C y D según corresponda).

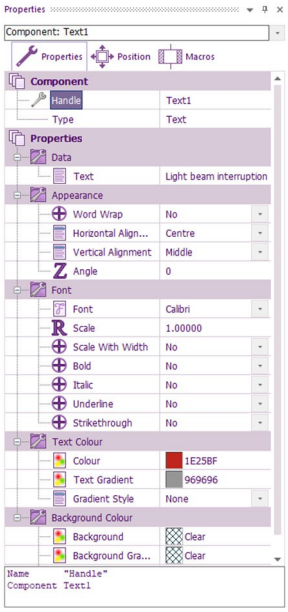
Añadir la matriz de LED

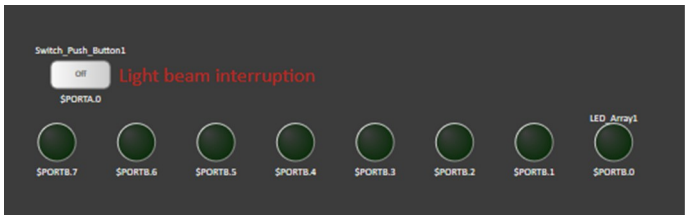
- Haga clic una vez en el cuadro "Salidas" y seleccione el icono "Matriz de LEDs" .Colócalo en el Panel de Sistema moviendo el cursor sobre él y haciendo clic y arrastrándolo hasta su posición.
- Cambia el valor de la propiedad 'Count'a '8'para establecer el número de LEDs en el array.
- Haz clic en la propiedad "Conexiones" del panel "Propiedades". Selecciona 'PORTB'en el menú desplegable para conectar los LEDs a los pines del puerto B.
- Puedes cambiar el color de la matriz de LED en la sección "Colores".

Añadir el interruptor

- Un único pulsador representará el sensor del haz luminoso.
- Seleccione **Pulsador Interruptor [2D]** en **Librerías de Componentes > Entradas**
- Añádalo o arrástrelo al Panel 2D.
- En la sección "Conexiones" del panel "Propiedades", compruebe que la propiedad "Conexión" del conmutador es "$PORTA.0", es decir, que el conmutador está conectado al puerto A bit 0.
- Seleccione Texto [2D] en **Bibliotecas de componentes > Creación > Primitivas (2d)**
- Haga clic en la propiedad Texto del panel Propiedades y sustituya el texto por defecto por "Interrupción del haz de luz".
- Para ajustar el tamaño del texto, haz clic en la pestaña "Posición" y cambia los valores de "Anchura" y "Altura" en la sección "Tamaño del mundo". Mueve el texto a una posición adecuada junto al interruptor.

Ahora debería tener un proyecto parecido a este:





En lugar de añadir texto, el texto archivado dentro del interruptor se puede cambiar de Switch_Push_Button1 a Interrupción del haz de luz.

Cambie la opción Etiqueta del componente de Igual que el asa a Personalizada y, a continuación, edite el campo Texto de la etiqueta de usuario.

Simulación del programa

- Ahora ejecute la simulación pulsando el botón Ejecutar .
- Aparece la ventana 'Depurador de simulación' - ciérrela ya que no es necesaria.
- Sitúa el cursor sobre el interruptor y haz el clic más breve que puedas.

Lo que ocurra dependerá de lo rápido que hagas clic y de lo rápido que funcione el PC.

Queremos que sólo se encienda el LED 'B0', para mostrar un total de 1 oveja. El programa se ejecuta a alta velocidad, sin embargo, y por lo que mantiene el ciclo a través de la "Entrada" y "Cálculo" pasos. Como resultado, antes de que tenga tiempo de soltar el pulsador, el total se habrá incrementado (aumentado en uno) varias veces. Este problema se analiza en la siguiente sección.

La solución: Añadir un retardo

El problema es que el programa va demasiado rápido.

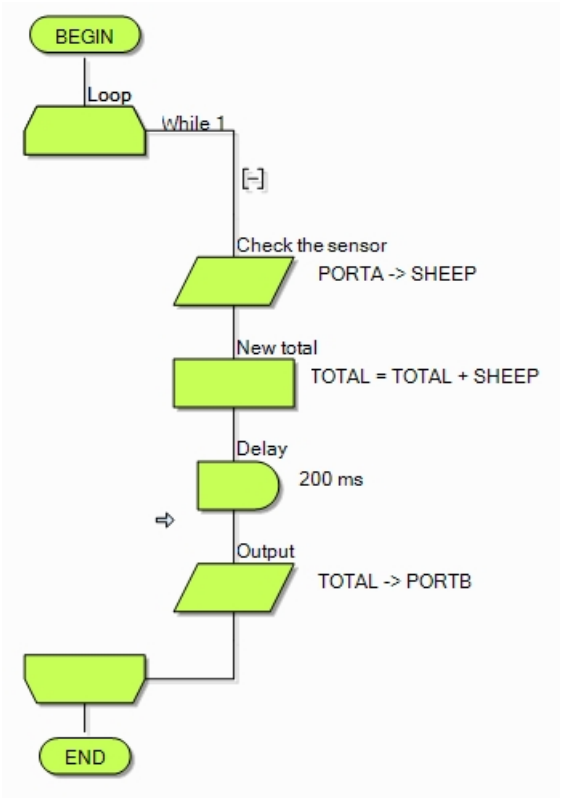
Antes de que tengamos tiempo de soltar el interruptor, el programa se ha ejecutado varias veces, sumando una al total cada vez

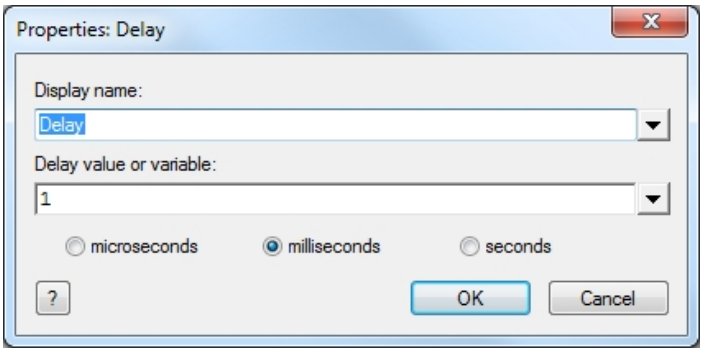
tiempo.

Tenemos que ralentizarlo añadiendo un dela.

- Sitúe el cursor sobre el icono 'Retraso'.
- Arrástrelo al área de trabajo principal y suéltelo entre los iconos de Cálculo y Salida.

El diagrama de flujo debería tener ahora este aspecto:





- Haga doble clic en el icono "Retraso" para abrir el cuadro de diálogo "Propiedades".
- Cambia el valor de la casilla "Valor o variable de retardo:" a "200" y haz clic en el botón "Aceptar". Esto provoca un retardo de 200 milisegundos (0,2 segundos) cuando se activa el icono 'Retardo'. En otras palabras, el sistema se queda ahí sin hacer nada durante 0,2 segundos.
- Ahora ejecuta de nuevo la simulación. Siempre que no lo mantengas pulsado demasiado tiempo, deberías ver que la matriz de LEDs muestra un incremento de 1 cada vez que pulsas el interruptor.
- El programa funciona ahora satisfactoriamente, siempre que las ovejas se precipiten a través del haz de luz en menos de 0,2 segundos. Se podría aumentar el retardo para que las ovejas fueran más lentas.

Nota: Este programa muestra el número total de ovejas en formato binario.

Ejemplo 3: La pantalla LCD

Flowcode viene con una serie de componentes que añaden subsistemas de uso común a Flowcode, como la pantalla LCD, la pantalla de 7 segmentos y los dispositivos de entradas analógicas.

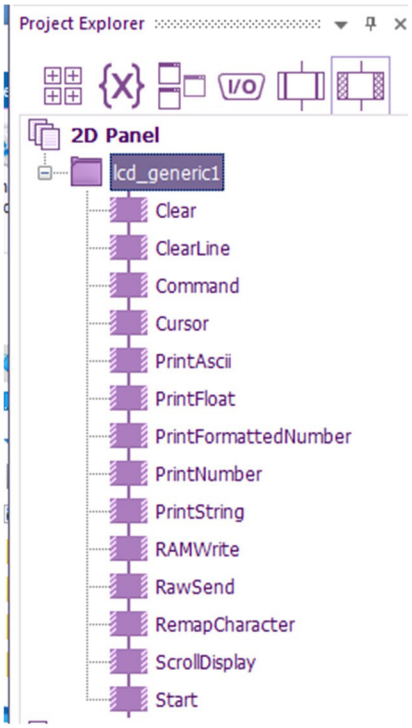
En este artículo analizaremos la pantalla LCD, el subsistema básico de visualización de texto en toda una serie de dispositivos electrónicos, desde calculadoras hasta teléfonos móviles. Puede mostrar texto o números en una o varias filas de la pantalla.

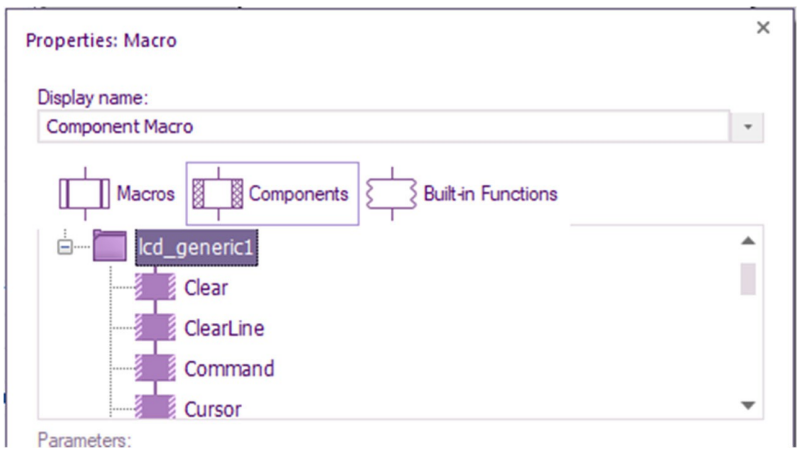
En la mayoría de los lenguajes de programación, la pantalla LCD e una de las últimas cosas que se aprenden, ya que es un dispositiv bastante complicado de programar.
Sin embargo, Flowcode se encarga de las complejidades, haciendo que el LCD fácil de usar. La pantalla LCD a la que se hace referencia aquí es la que se utiliza en la placa E-Blocks Combo y en la pantalla LCD: una pantalla de dos filas y dieciséis caracteres.

Añadir el componente LCD

Antes de poder utilizar la pantalla LCD, es necesario añadir un componente LCD a un panel Flowcode.

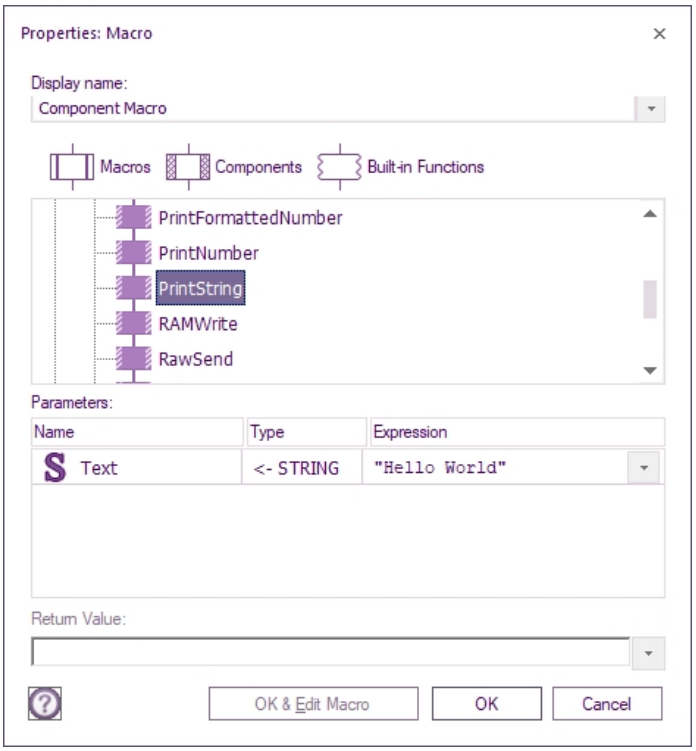
- Seleccione el componente **LCD (Genérico, Configurable) [2D]** en **Bibliotecas de componentes > Pantallas** y añádalo al **Panel 2D**. Aparecerá una pantalla LCD en el panel.
- Dentro del explorador de proyectos **Ver > Explorador de proyectos** identifica el componente que acabas de seleccionar. Por defecto, la pantalla LCD se añade al puerto B. Puedes cambiar esto, pero lo mantendremos en el puerto B.
- Hemos añadido una pantalla LCD al programa. ¿Está listo para usar? ¿Cómo se utiliza?
- La pantalla LCD requiere cinco conexiones. Muestra letras y números transmitidos como datos serie en este bus de cinco hilos.
- Las técnicas implicadas van más allá de este tutorial. Afortunadamente, Flowcode tiene algunas rutinas integradas que se encargan de las complejidades.
- Arrastre el icono "Iniciar macro componente" al diagrama de flujo desde el explorador de proyectos.
- Como alternativa, haga clic con el botón derecho del ratón en el diagrama de flujo donde desee colocar la macro componente y seleccione Añadir > Macro componente. abra el cuadro de diálogo de la macro haciendo doble clic sobre ella.
- Expanda el lcd_generic y haga clic en el símbolo + Ahora desplácese por la sección 'LCD' en 'Componentes' y seleccione la macro llamada 'Inicio'. Esto inicia la LCD, limpia la pantalla y la prepara para la acción. Examinaremos más macros de LCD en las próximas secciones, pero por ahora recorra las macros disponibles y eche un vistazo rápido a cada una.



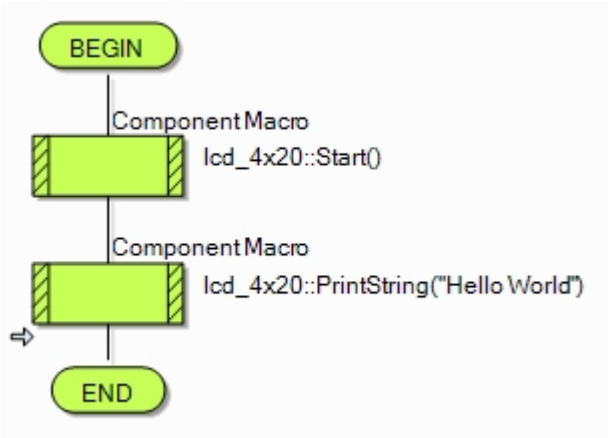


Escribir mensajes

Para visualizar texto en la pantalla LCD, basta con escribirlo.



- Añada otra macro componente al diagrama de flujo y abra el cuadro de diálogo de la macro.
- Seleccione la macro LCD llamada 'PrintString'. Esto requiere un único parámetro (elemento de datos), 'Texto', - el texto a imprimir.
- Escriba el texto en la casilla de parámetros entre comillas, por ejemplo: "Hola Mundo".
- Ejecute el programa y el texto se enviará a la pantalla LCD.



Otras funciones LCD

Hay otras funciones útiles en la lista de macros de la pantalla LCD:

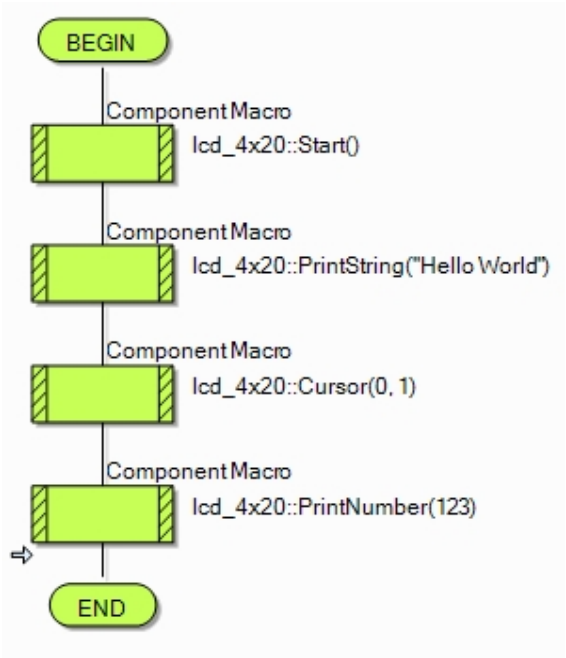
- Borrar" - Borra la pantalla y restablece la posición del cursor (donde se imprimirá a continuación) a "0,0", es decir, arriba a la izquierda.
- 'Cursor'- Mueve el cursor a la posición especificada. Los dos parámetros, 'X'e 'Y'seleccionan las posiciones horizontal y vertical de la celda respectivamente. '0,0'es la celda superior izquierda, '0,1'la primera celda de la segunda línea, '3,2'la cuarta celda de la tercera línea ... .
- PrintNumber'Funciona como 'PrintString'pero imprime un número en lugar de una cadena. Se puede utilizar con variables o con números reales.

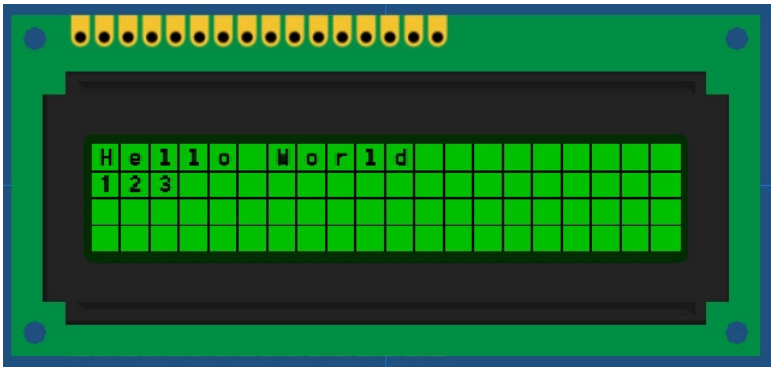
Uso de PrintNumber - un ejemplo:

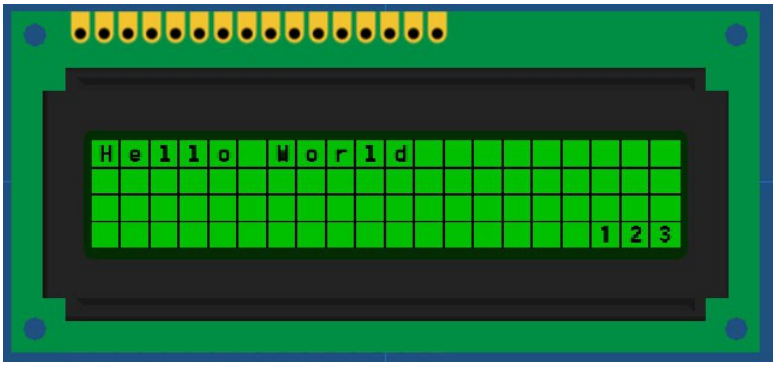
En total, añadiremos cuatro **macros de componentes** al diagrama de flujo.

- A la primera **Macro Componente** añada **Inicio**.
- En el segundo seleccione **PrintString** y añada **"Hola Mundo"** (entre comillas).
- Al tercero selecciona **Cursor** y añade **0,1** a los parámetros.
- A la cuarta seleccione **PrintNumber** con el valor del parámetro como **123.**
- seleccione 'PrintString'y añada "Hola Mundo" (entre comillas) como parámetro;
- Haga clic en "Ejecutar" para simular el programa.

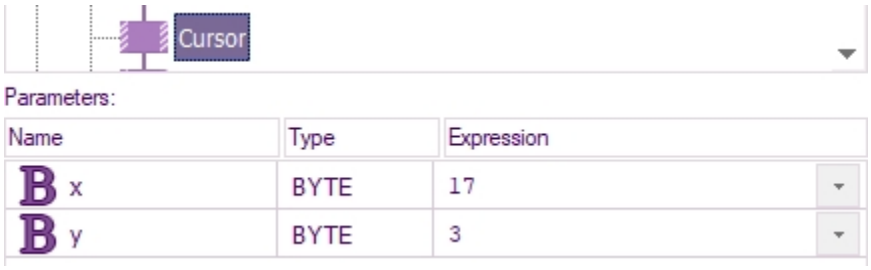
Debería ver un resultado similar al que se muestra a continuación:







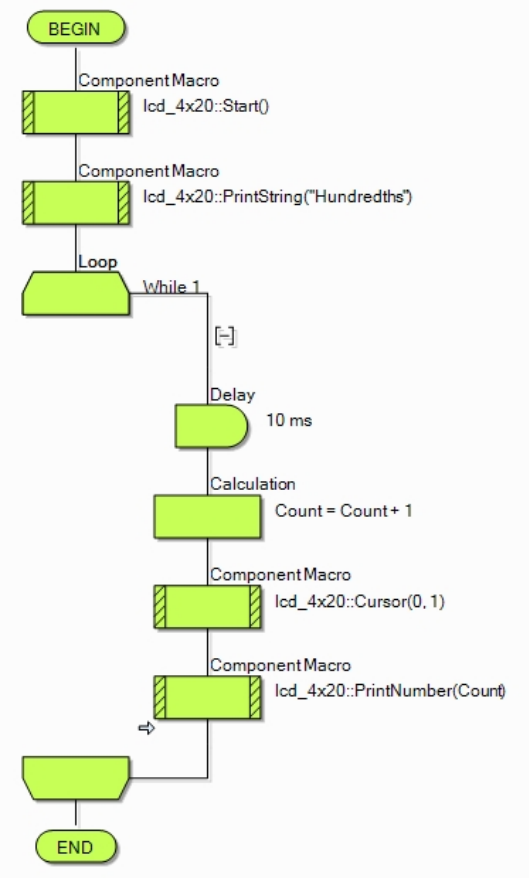
CONSEJO: Pruebe a cambiar los valores de los parámetros **del Cursor** y vea dónde se imprimen los números. El valor 'y' tiene que estar entre 0 y 3 y el valor 'x' tiene que estar entre 0 y 19. (entre 3 y 17 para ver las tres figuras 1 y 2 y 3).

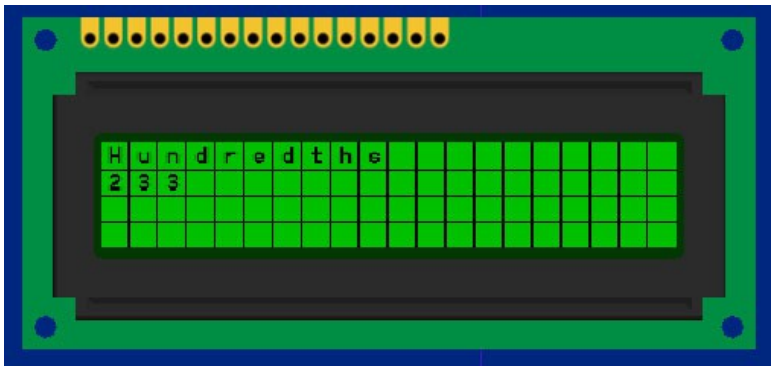


Ejemplo 4: un cronómetro

Este ejemplo utiliza el ejemplo 3 (Uso de PrintNumber) como punto de partida.

- Amplíe el programa del ejemplo anterior (**Utilizar PrintNumber**) arrastrando un icono de **bucle** debajo del icono **Macro del componente PrintString**.
- Cambie el texto de la macro 'PrintString'Component por "Centésimas:" (entre comillas).
- Arrastre un icono de "Cálculo" al bucle.
- Crear una variable llamada 'Count'de tipo 'Int'. (valor inicial 0)
- Haga doble clic en el icono 'Cálculo'. En el cuadro de texto 'Cálculos:'escriba "Recuento = Recuento + 1". Esto añadirá 1 al valor de la variable recuento cada vez que se ejecute el icono.
- A continuación, arrastre otro 'Component Macro' al Bucle.
- Haga doble clic en 'Component Macro' y busque 'Cursor' bajo las macros 'LCD'.
- Introduzca '0,1'como parámetros para situar el cursor en el primer carácter de la segunda línea.
- A continuación, arrastre otra "Macro componente" al área de trabajo.
- Seleccione 'ImprimirNúmero'e introduzca 'Cantidad'como parámetro.
- Ahora, arrastra un icono de "Retardo" al diagrama de flujo y establece el retardo en 10 ms (que equivale a una centésima de segundo).
- Perfeccione el programa haciendo clic en cada icono e introduciendo comentarios sobre lo que hace el icono. Puede parecer mucho esfuerzo, pero te ahorrará tiempo después, ya que tu programa será más fácil de seguir.
- Ejecute el programa. Ahora ha creado un contador que contará (aproximadamente) el tiempo transcurrido en centésimas de segundo.





CONSEJO: Puede afinar el programa haciendo clic en cada icono e introduciendo comentarios para describir lo que hace el icono.

Puede parecer mucho esfuerzo, pero puede ayudar con programas más complejos.

Ejemplo 5 - Utilización de números binarios - Sumador binario

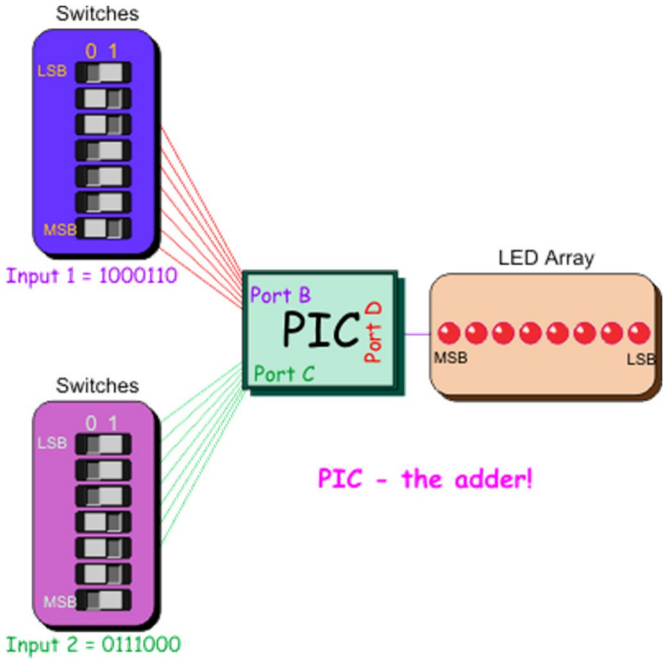
En esta sección construyes un sistema que hace que el microcontrolador sume dos números.

La forma más sencilla de introducir un número binario es utilizar un conjunto de interruptores conectados al puerto de entrada.

Para introducir dos números, necesitamos dos juegos de interruptores y dos puertos de entrada.

Para ver el resultado del cálculo, utilizaremos una matriz de LEDs, conectada al puerto de salida.

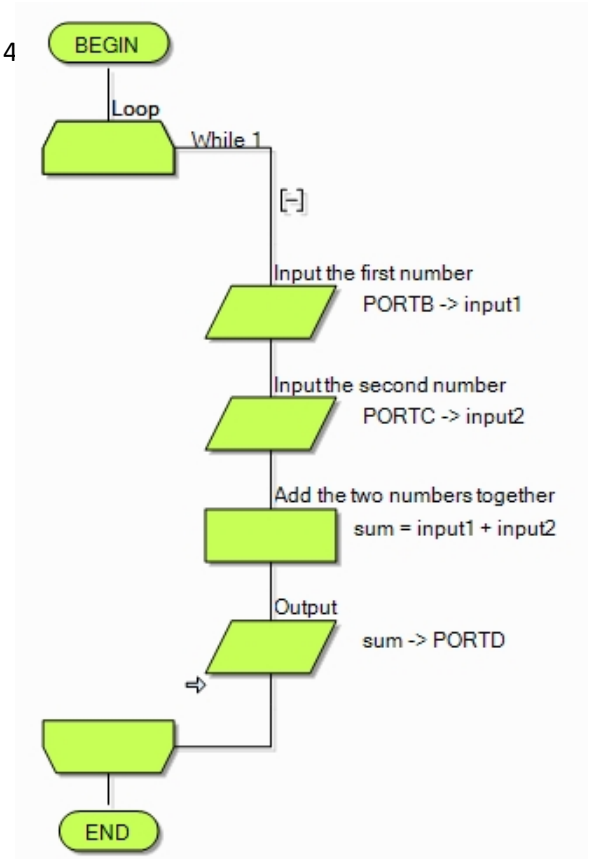
Necesitamos un chip PIC con tres puertos.



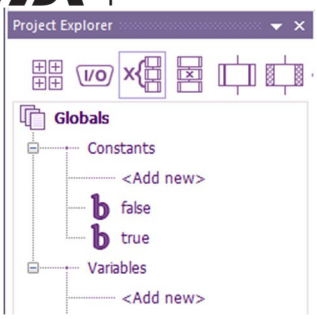
Configuración del organigrama

- Ejecute 'Flowcode' e inicie un nuevo diagrama de flujo.
- Esta vez nos fijamos en este cuadro de diálogo:
- Necesitamos un PIC con al menos tres puertos.
- Cargue el BL00011 o el 16F18877 PIC como hizo en la página 4
- Haga clic y arrastre un icono de bucle entre los cuadros "INICIO" y "FIN".
- Haz clic y arrastra un icono de entrada y colócalo entre los extremos del bucle.
- Haz clic y arrastra un segundo icono "Entrada" y colócalo entre los extremos del bucle.
- Haga clic y arrastre un icono de Salida y suéltelo justo debajo de los cuadros de 'Entrada'.
- Haga clic y arrastre un icono de Cálculo y colóquelo entre el segundo icono de Entrada y el icono de Salida.
- Su diagrama de flujo debería tener ahora este aspecto:

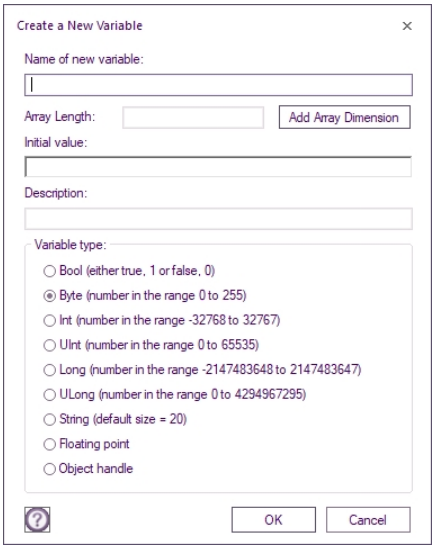
(a la imagen de ejemplo se le han añadido descripciones y variables).



Creación de las variables

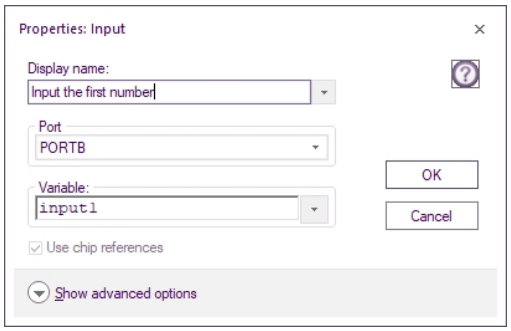


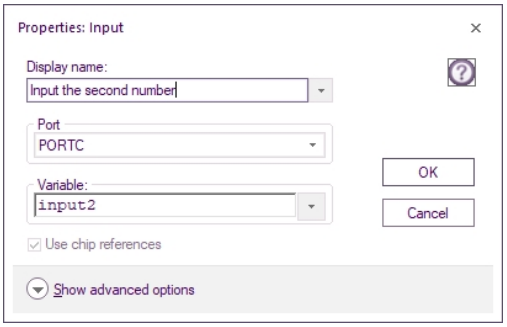
- Haga clic en "Ver" en la barra de menús y asegúrese de que la opción "Explorador de proyectos" está seleccionada (Ver > Explorador de proyectos).
- Haga clic en el botón "Globales" en la parte superior del panel Explorador del proyecto. Vamos a crear tres variables, llamadas 'input1', 'input2'ny 'sum'. Las dos primeras almacenan los números introducidos por los interruptores. La variable 'suma' almacena el resultado de sumarlos.
- Pase el ratón por encima de "Variables" en el panel "Explorador de proyectos" y haga clic en el icono que aparece.
- Haz clic en "Añadir nueva" y aparecerá el cuadro de diálogo "Crear una nueva variable". Escribe el nombre "input1", y haz clic en el botón 'OK'- deja el tipo de variable como 'Byte'.
- Crear variables, 'input2'y 'sum'de la misma manera.



Configuración de las entradas

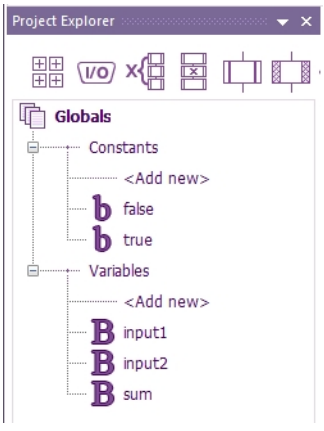
- Haga clic con el botón derecho en el icono superior "Entrada" y seleccione "Propiedades". Aparecerá el cuadro de diálogo "Propiedades: Entrada".
- Haga doble clic en la palabra "Entrada" del cuadro "Nombre para mostrar:" para resaltarla.
- Escriba "Introducir el primer número" para sustituirlo. Aparecerá junto al icono "Introducir" en el diagrama de flujo. (Añadir etiquetas de este tipo ayuda a los usuarios a entender lo que está ocurriendo).
- Haga clic en las flechas situadas junto al cuadro de variables para abrir el "Gestor de variables". Aquí aparecen las tres variables que acabas de crear.
- Haga doble clic en 'input1'para utilizar esta variable en el cuadro de entrada.
- De nuevo en el cuadro de diálogo "Propiedades de entrada", haz clic en la flecha hacia abajo situada al final de la ventana del puerto y selecciona "PORTB" en lugar de "PORTA".
- Haga clic en "Aceptar" para cerrar el cuadro de diálogo.
- Haga doble clic en el segundo icono 'Entrada'. (una forma más rápida de abrir el cuadro de 'Propiedades').





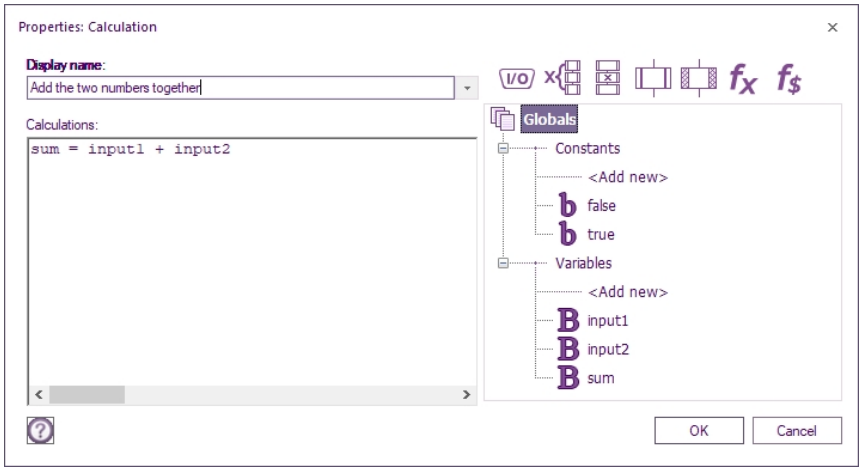
Configura esta entrada para:

- mostrar la etiqueta "Introduzca el segundo número";
- utilizar la variable 'input2';
- utilice "PORTC".
- A continuación, cierra el cuadro de diálogo pulsando el botón "Aceptar".



Para los usuarios de Arduino, estos dos puertos deberán configurarse de la siguiente manera: Entrada 1 ajustada a PORTC (para utilizar los interruptores del Puerto A en la placa Combo). Entrada 2 ajustada a PORTD (para usar los interruptores del Puerto B en la placa Combo).

Establecer el cálculo



- Haga doble clic en el icono Cálculo para abrir el cuadro de diálogo Propiedades.
- Cambia el 'Nombre a mostrar:'por 'Suma los dos números'.
- En la casilla 'Cálculos:'inserte:

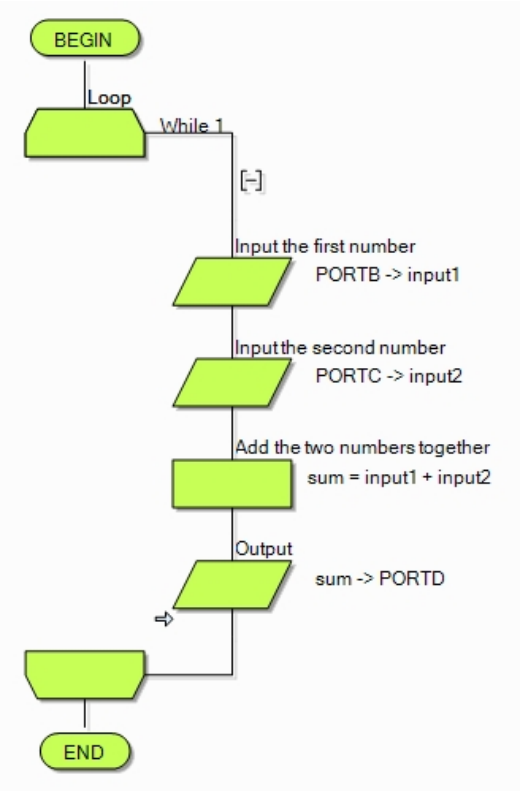
  suma = entrada1 + entrada2

(Puede escribirlo directamente o arrastrar las variables desde la ventana de la derecha y luego insertar los signos "=" y "+" en el lugar correcto).

- A continuación, haga clic en el botón "Aceptar" para cerrar el cuadro de diálogo.

Configurar la salida

- Haga doble clic en el icono "Salida" para abrir el cuadro de diálogo "Propiedades" de la salida.
- Haga clic en la flecha situada junto a la casilla "Variable:".
- Haz doble clic en 'suma'para insertarla en la casilla.
- De nuevo en el cuadro de diálogo 'Propiedades' de la salida: cambia el puerto utilizado a 'PORTD'. *(Arduino PORTB)*
- Haga clic en "Aceptar" para cerrar el cuadro de diálogo. El diagrama de flujo debería tener ahora este aspecto:



Añadir una matriz de LED

- Haz clic en la pestaña "Salidas" y selecciona "Matriz de LEDs".
- Colóquelo en el centro del Panel de Sistema moviendo el cursor sobre el componente y haciendo clic y arrastrándolo hasta su posición (o haciendo clic con el botón derecho y seleccionando "Centrar todos los objetos").
- Haga clic en la propiedad "Count" de la sección "Simulation" del panel Properties y cambie el número de LEDs a siete.
- Haz clic junto a la propiedad 'Puerto' y selecciona 'PORTD' en el menú desplegable para conectar los LEDs a los pines del puerto D. *(Arduino PORTB)*
- Cambia el color de la matriz de LEDs a rojo (0000FF), cambiando la propiedad 'LED 0'mientras la propiedad 'Mismo Color'está en 'Sí'.

Añadir los interruptores

Se utilizan dos juegos de conmutadores, uno para cada número binario. El puerto de salida sólo tiene ocho bits. El mayor número que puede emitir es 1111 1111, (= 255 en decimal). Vamos a limitarnos a introducir números de siete bits, lo que significa que el mayor número que podemos introducir es 111 1111, (= 127 en decimal). Si utilizáramos números mayores, desbordaríamos la capacidad de la salida.

- Haga clic en la pestaña "Entradas", seleccione "Matriz de interruptores" y arrástrela al panel del sistema, encima de la matriz de LED.
- Abre el panel de propiedades del switch array. Conéctalo al puerto B, utilizando el juntoa la propiedad 'Puerto' para abrir el menú desplegable.*(Arduino PORTC)*
- Añada una segunda matriz de interruptores al panel del sistema del mismo modo. Colócalo debajo del 'Array LED' y conéctalo a 'PORTC'.

(Arduino PORTD)

Simulación lenta

Como se ha descrito anteriormente, Flowcode le permite avanzar por el diagrama de flujo paso a paso/icono a icono, para ver el efecto de cada uno en las variables y en la salida.

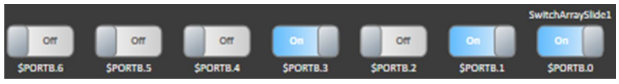
Hay tres formas de simular el programa paso a paso:

- Haga clic en **Ir** en la barra de herramientas Depurar y en el botón **Step Into (Depurar > Step Into)**
- Pulse la tecla de función F8 del teclado.
- Haz clic en el botón "Step Into" de la barra de herramientas principal en la

sección de simulación. Haz una de estas cosas

Ocurren varias cosas:

- aparece un rectángulo rojo alrededor del icono 'BEGIN', indicando que se trata del paso actual;
- aparece la ventana 'Simulation debugger'- que contiene 'Variables'y 'Macro Calls';
- la sección "Variables" enumera las tres variables que ha definido para este programa y muestra sus valores actuales - todo a cero en este momento.

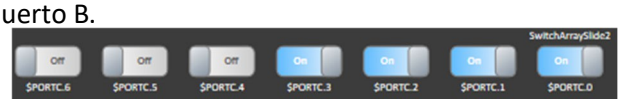


Ignora la sección 'Macro Calls' por el momento.

Ahora configura dos números en los componentes

del interruptor.

- Sitúe el cursor sobre la caja de conmutación conectada al puerto B.
- Haz clic en los interruptores B0, B1 y B3 para activarlos.
- Los interruptores tienen ahora este aspecto:



- Has configurado el número binario 000 1011 (= once en decimal.)
- (El conmutador 'B6'da el bit más significativo y 'B0'el bit menos significativo).
- Configure el número 000 1111 (quince) en los conmutadores conectados al puerto C.
- Ahora 'Step Into'al siguiente icono en el programa, por ejemplo, pulsando F8 una vez más.
- El rectángulo rojo pasa al siguiente icono, el de 'Bucle', pero no ocurre nada más.
- Pulse F8 una vez más. El rectángulo rojo pasa al primer icono de Entrada.
- Vuelve a pulsar F8 y el cuadro "Variables" mostrará que la variable "input1" contiene ahora once, el resultado de la instrucción "Input" que acabamos de ejecutar.
- Pulsa F8 de nuevo y la sección "Variables" mostrará que "input2" contiene ahora quince.
- Vuelva a pulsar F8 y se realizará el cálculo. La variable 'suma'almacena el resultado.
- Pulse F8 de nuevo. El valor almacenado en 'sum'se transfiere al

LEDarray. Se ve como:

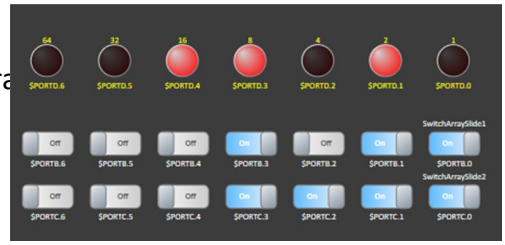
Leyendo desde el bit más significativo ('D7') hasta el menos

sig

nificativo ('D0'), la matriz de LED muestra el número 0001 1010. En decimal, es el número 26. No hay sorpresas.

Repite el mismo procedimiento utilizando números diferentes y avanza por el programa para comprobar cuál es la suma de los números.

CONSEJO: Explore la posibilidad de añadir gráficos a su calculadora binaria para facilitar su lectura. **Bibliotecas de componentes > Creación** para añadir dígitos encima de tus LED.



Ejemplo 6. Lógica binaria en el control.

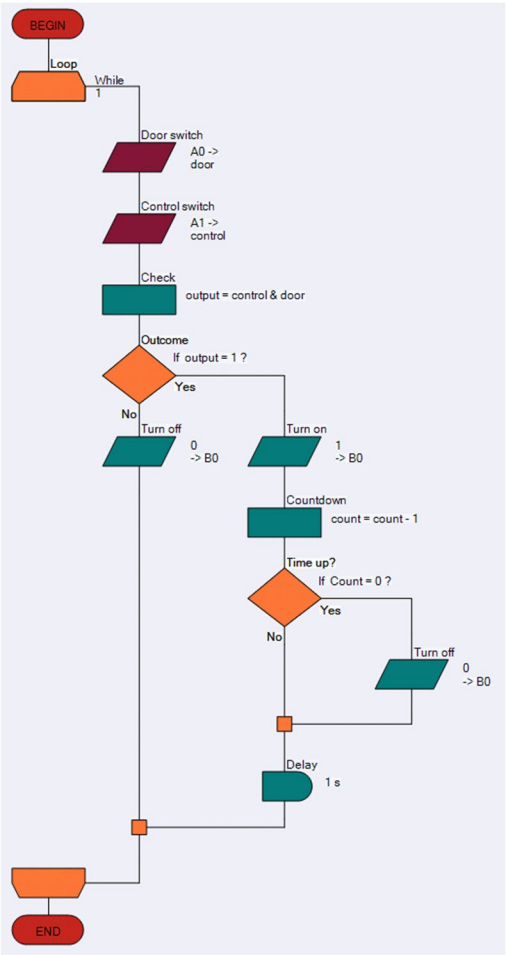
Los sistemas electrónicos pueden tomar decisiones.

A menudo son del tipo "Si esto Y esto es cierto, entonces..." o "Si esto O esto es cierto, entonces...". Se basan en combinaciones específicas de circunstancias para emprender una determinada acción.

Son ejemplos de uso de la lógica binaria. La respuesta a la pregunta "Si..." es "Sí" / "No", o "Verdadero" / "Falso", es decir, una de dos posibilidades (una solución binaria). Esta respuesta puede expresarse como un 0 lógico o un 1 lógico y electrónicamente mediante una tensión alta o una tensión baja.

Existe una clase de componentes electrónicos digitales, llamados puertas lógicas, que toman exactamente estas decisiones. Las entradas y salidas son 0 o 1 lógicos.

Podemos programar Flowcode para que tome exactamente las misma



6A. Control de un horno microondas

Por razones de seguridad, un horno microondas tiene un sensor de puerta para asegurarse de que el generador de microondas no funcione si la puerta está abierta. Dicho de otro modo, el generador funciona si la puerta está cerrada Y se pulsa uno de los interruptores de control de la calefacción. Podemos incluir esta condición en un programa Flowcode.

Configuración del organigrama

Inicie Flowcode con un nuevo diagrama de flujo.

Crea el diagrama de flujo que se muestra al lado. Utiliza:

- un icono de bucle
- dos iconos de entrada
- tres iconos de salida
- dos iconos de decisión
- dos iconos de cálculo
- un icono de

retraso. Crea cuatro

variables:

- 'puerta'(para almacenar el estado del interruptor de la puerta).
- 'control'(para almacenar el estado del interruptor de control de encendido/apagado)
- salida" (para controlar si el microondas se enciende o no)
- 'count'(para controlar cuántas veces se ha producido el retardo de 1s. Dale un valor inicial de diez, para que el horno microondas funcione durante 9s).
- Utiliza la configuración por defecto para el icono de bucle.
- Configurar un icono de entrada para almacenar el estado del interruptor de la puerta (en el puerto A bit 0) en la variable 'puerta'.

- Configura el otro icono de entrada para almacenar el estado del interruptor de control (en el puerto A bit 1) en la variable 'control'.
- El icono de cálculo superior comprueba si se ha pulsado la puerta Y el interruptor de control.

  Configúrelo mediante la ecuación salida = mando & puerta.

  La & significa la operación AND.

  El resultado de esta operación (0 ó 1) se almacena en la variable 'salida'.
- El icono de decisión superior comprueba el valor

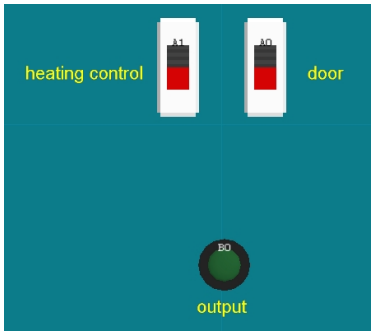
  almacenado en 'output'. (If output? es la abreviatura de

  If output=1?)

  Configura este icono de decisión.
- Cuando el resultado del cálculo es 0, el programa sigue la ruta "No" del icono de decisión y se ejecuta el icono de salida de la izquierda. Esto envía un 0 lógico al LED, asegurando que éste (y el generador de microondas) se apague.
- Cuando el resultado del cálculo es 1, el programa sigue la ruta 'Sí'. El icono de salida 'Encender' envía un 1 lógico al LED encendiéndolo.

  Configure ambos iconos de salida.
- El icono de cálculo inferior reduce en uno el número almacenado en la variable

  'count'. Configúralo utilizando la ecuación count = count - 1
- El valor inicial de 'count'es diez. Siempre que el número almacenado en 'count'no haya llegado a cero, el programa sigue la ruta 'No'. Eventualmente, después de hacer un bucle suficientes veces, el número almacenado se reduce a cero. El programa sigue entonces la ruta 'Sí' y ejecuta el icono de salida 'Apagar', que se configura de la misma manera que el otro icono 'Apagar', para apagar el generador de microondas.



- Añada una matriz de conmutadores al Panel de Sistema. Configúralo para que sólo tenga dos conmutadores, uno conectado al puerto A, bit 0 y el otro al puerto A, bit 1.
- Añade un LED conectado al puerto B, bit 0 para representar el generador de microondas.
- Añada etiquetas al Panel de Sistema para identificar los componentes. Colóquelas utilizando las coordenadas mundiales en la pestaña Posición de las propiedades de la etiqueta.
- Ahora simule el programa paso a paso, utilizando repetidamente la tecla de función F8.
- Compruebe lo que ocurre para diferentes combinaciones de estados del interruptor e interprételo en términos del comportamiento del horno microondas. ¿Qué ocurre, por ejemplo, si se abre la puerta mientras el generador de microondas está en funcionamiento?

Para Arduino, los puertos deben ajustarse a PORTC y PORTD (equivalentes a A y B en la placa Combo).

Ejemplo 6. Lógica binaria en el control.

6B Control de la luz interior de un coche.

La luz interior de un coche puede controlarse mediante otra ecuación lógica booleana.

Para simplificar, consideremos un coche de dos puertas con el siguiente

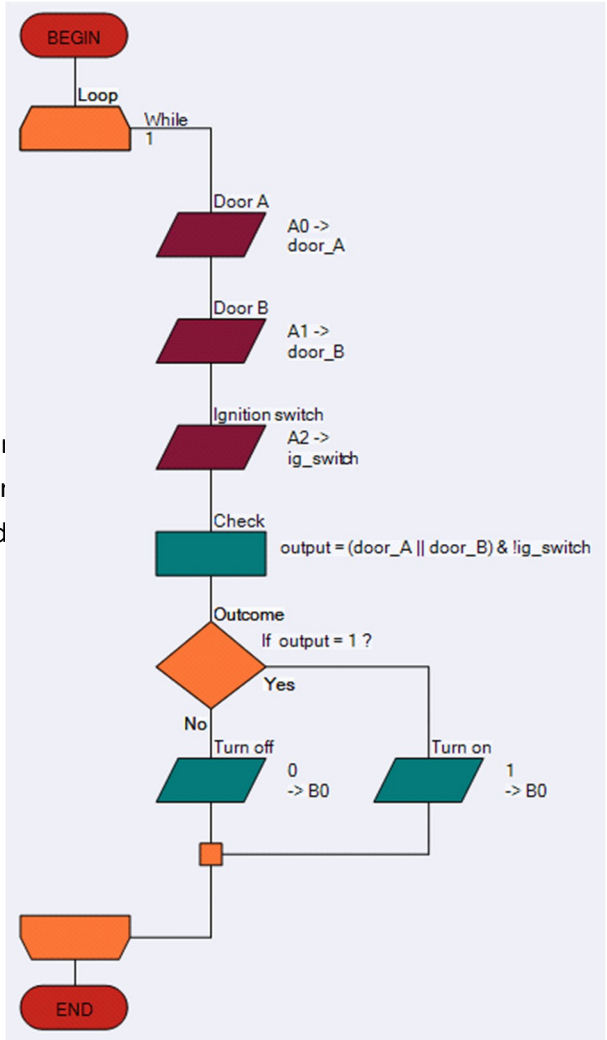
La luz interior se enciende cuando se abre una puerta (A) O la otra (B) y permanece encendida hasta que se gira la llave de contacto (C). En lenguaje booleano, decimos que la luz está encendida si (A O B) Y NO C es cierto.

Una vez más, podemos incorporar esta condición a un programa Flowcode.

Configuración del organigrama

Inicie Flowcode y cree un nuevo diagrama de flujo. Cree el diagrama de flujo que se muestra al lado, utilizando:

- un icono de bucle.
- tres iconos de entrada.
- dos iconos de salida.
- un icono de decisión.
- un icono de cálculo.



Crea cuatro variables:

- puerta_A (para memorizar el estado del interruptor de la puer
- puerta_B (para memorizar el estado del interruptor de la puer
- ig_switch (para almacenar el estado del interruptor de encend
- (para controlar si la luz interior se enciende o no).

- Utiliza la configuración por defecto para el icono de bucle.
- Configure un icono de entrada para almacenar el estado del interruptor de la puerta A (puerto A bit 0), en la variable 'puerta_A'.
- Configure un icono de entrada para almacenar el estado del interruptor de la puerta B (puerto A bit 1) en la variable 'door_B'.
- Configure el otro icono de entrada para almacenar el estado del interruptor de encendido (puerto A bit 2) en la variable 'ig_switch'. El icono de cálculo comprueba si alguna de las puertas se ha abierto Y el interruptor de encendido NO está encendido.
- Configúralo utilizando la ecuación salida = (puerta_A ||puerta_B) & !ig_switch
- || significa la operación OR y ! la operación NOT. El resultado del cálculo se almacena en la variable 'output'.

(Para los usuarios de Arduino, utilice los puertos C y D según corresponda).

- El icono de decisión comprueba el valor almacenado en 'salida'.
- Configura este icono de decisión.
- Cuando el resultado del cálculo es 0, el programa sigue la ruta 'No' del icono de decisión y se ejecuta el icono de salida 'Apagar', asegurando el apagado de la luz.
- Cuando el resultado del cálculo es 1, el programa sigue la ruta 'Sí'. El icono de salida 'Encender' envía un 1 lógico al LED encendiéndolo.
- Configure ambos iconos de salida.
- Añada una matriz de conmutadores al Panel de Sistema. Configúralo para que tenga tres conmutadores, uno conectado al puerto A, bit 0, otro al puerto A, bit 1 y otro al puerto A, bit 2.
- Añade un LED conectado al puerto B, bit 0 para representar la luz interior del coche.
- Añada etiquetas al panel del sistema para identificar los componentes y colóquelos como se muestra en el diagrama. **(Bibliotecas de componentes > Creación)**

Ahora simule el programa paso a paso, utilizando repetidamente la tecla de función F8.

Compruebe lo que ocurre para diferentes combinaciones de puertas abiertas y estados de la llave de contacto. Interpreta el comportamiento en función del comportamiento de la luz interior. ¿Qué ocurre, por ejemplo, si se abre la puerta y se cierra poco después? ¿Es correcto este comportamiento?

Sección 6: Programación ejercicios

Los **Ejercicios de programación** se presentan aquí como tareas flexibles que pueden desarrollarse posteriormente.

Si lo desea, las pequeñas tareas individuales pueden convertirse en proyectos a mayor escala. Pruebe las ideas, póngalas a prueba, experimente, desarrolle sus habilidades y vea lo que es capaz de crear.

El objetivo de los ejercicios es desarrollar la experiencia en el uso de Flowcode y, en el proceso, desarrollar la comprensión de la terminología y las técnicas de programación que abarca.

Los programas pueden probarse simulándolos en Flowcode, pero también pueden descargarse a un microcontrolador y probarse en hardware. En general, se supone que el programador utiliza un Microchip PIC MCU, aunque los ejercicios son igualmente aplicables a otros microcontroladores.

La sección termina con otros **retos.** Estos son aún más abiertos y solo contienen una breve especificación.

Introducción

Este ejercicio configura Flowcode para enviar señales digitales específicas a la matriz de LEDs.

Fondo

- Sección 1 - Introducción a los microcontroladores.
- Sección 2 - Utilización de los bloques electrónicos.
- Sección 4 - Flowcode Primer Proyecto. Añadir salidas digitales - Encender el LED
- Flowcode Wiki - Uso de máscaras.

Objetivos

- Cambia el nivel lógico de un solo pin de un puerto.
- Envía diferentes códigos de 8 bits al puerto de un microcontrolador.
- Configurar un icono de **salida**.
- Utiliza código **binario**.
- Manipula los niveles **lógicos** de salida.
- Utiliza LEDs para visualizar una **salida**.

Tareas

1. Crea un programa Flowcode:

- añade un único icono de salida, configurado para encender todos los LEDs del puerto B y ejecuta la simulación;
- modifica los parámetros para encender sólo los LED impares y ejecuta la simulación;
- haga lo mismo pero sólo para los LED pares;
- haga lo mismo pero sólo para los bits altos (4 a 7) del puerto B.

modifica este programa a ver si puedes:

- repitiendo estos cuatro pasos utilizando la numeración hexadecimal en lugar de la decimal;
- encender sólo el LED del bit 7, enviando un valor de 8 bits al puerto;
- encender sólo el LED del bit 7, utilizando el método de salida 'bit único';
- encender sólo el LED del bit 7, utilizando el método de salida 'masking'.

2. Escribe un programa que utilice al menos veinte iconos de **salida** para escribir diferentes valores en el puerto B, uno tras otro. Utilice en este ejercicio los cuatro métodos: hexadecimal, decimal, bit a bit y enmascaramiento. Simule el programa y revise los resultados. (Guarda el programa y descárgalo en el microcontrolador).

CONSEJO: Reinicie el programa varias veces pulsando el botón Reset de la placa programadora.

Introducción

En este ejercicio, aprenderás cómo se utilizan los retardos para ralentizar el PIC. Los microcontroladores trabajan extremadamente rápido - un PIC puede ejecutar alrededor de 5.000.000 de instrucciones en ensamblador, cada segundo. A humano puede detectar y comprender sólo unas tres imágenes estables por segundo. Para que el PIC de alta velocidad pueda comunicarse con los humanos "lentos", a veces tenemos que ralentizarlo añadiendo instrucciones de retardo.

Fondo

- Sección 1 - Introducción a los microcontroladores.
- Sección 2 - Utilización de los bloques electrónicos.
- Flowcode Wiki - Propiedades del icono de bucle.

Objetivos

- Añade un retardo para ralentizar la ejecución de un programa.
- Cambia el intervalo de retardo.
- Configura un icono de retardo.
- Controla la velocidad de un microcontrolador.
- Utilizar un **osciloscopio** para cronometrar eventos .

Tareas

1. Comience abriendo el programa creado en el último ejercicio (Ejercicio 1).

- Añade iconos de retardo y configúralos para que los estados de salida se puedan ver cómodamente incluso a la velocidad del oscilador HS;
- guardar el programa y descargarlo al PIC probando el programa en las placas E-blocks.

Modificar la duración de los retardos provocados por los iconos de Retardo:

- empezar con un retraso de 1s;
- reduce progresivamente el retardo hasta que sea demasiado rápido para que tus ojos detecten los diferentes estados de salida;
- descargar el programa en el PIC cada vez y probarlo en E-blocks.
- utilice un osciloscopio para medir los retardos que configuró en Flowcode;
- hacer un dibujo detallado de la imagen del osciloscopio, con información completa sobre la tensión y la temporización y el tiempo de retardo utilizado en el programa Flowcode.

CONSEJO: No lo pruebe en modo de simulación, ya que los tiempos de simulación no siempre son precisos porque se ejecutan en un sistema operativo Windows y no en "tiempo real".

Introducción

Un punto de conexión, o instrucción 'goto', se utiliza a menudo para crear un bucle infinito - para repetir un conjunto de instrucciones una y otra vez. (Una mejor manera de hacer esto es utilizar una instrucción 'Loop').

La ventaja de un Punto de Conexión es que se puede utilizar para saltar de un bucle a un lugar determinado del programa. Se introduce la idea de modulación por ancho de pulsos (PWM) como medio para controlar el brillo de los LEDs.

Fondo

- Flowcode Wiki - Propiedades del icono de punto de conexión.
- Sección 1 - Introducción a los microcontroladores.
- Sección 2 - Utilización de los bloques electrónicos.
- Sección 4 - Flowcode Primer Proyecto. Añadir salidas digitales - Encender el LED.

Objetivos

- Utilice **los puntos de conexión** para introducir bifurcaciones incondicionales en un programa.
- Introducir **PWM** como medio para controlar el brillo de los LED.
- Crea un **bucle infinito**.
- Manipula los niveles lógicos de salida.
- Utiliza LEDs para visualizar una salida.

Tareas

1. Escribe un programa para ver si puedes:

- Utiliza los iconos de Retardo, Salida y Punto de Conexión para encender los LEDs pares e impares del puerto B, alternativamente encendidos y apagados, con un intervalo de 300ms entre ellos, en un bucle infinito;
- probar el programa primero 'paso a paso'y luego de forma continua en el simulador Flowcode.;
- utiliza los iconos de Retardo, Salida y Punto de Conexión para encender y apagar alternativamente los LEDs de nibble alto y nibble bajo del puerto B, con un intervalo de 300ms entre cada uno, en un bucle infinito;
- utiliza los iconos Retardo y Salida para encender y apagar todos los LEDs del puerto B con un intervalo de 500ms entre cada uno, en un bucle infinito;
- descarga el programa en el microcontrolador y pruébalo.

CONSEJO: Haga que los retardos en este programa sean muy cortos y que los tiempos de encendido y apagado sean asimétricos, (por ejemplo, encendido durante 8ms y apagado durante 12ms).

Se trata de un generador PWM por software. Cuando lo ejecutas, la intensidad de los LEDs es menor. Se encienden y apagan demasiado rápido para que nuestros ojos puedan observarlos. En su lugar, vemos el cambio de intensidad.

2. Escribe un programa que:

- enciende los LEDs de los cuatro bits más significativos, (MSB,) del puerto B y los mantiene encendidos;
- atenúa la intensidad de los LEDs en los cuatro bits menos significativos, (LSB,) del puerto B usando PWM, para crear una diferencia observable en intensidad entre los LEDs MSB y los LEDs LSB;
- utilice un osciloscopio para examinar la señal que controla uno de los cuatro LEDs LSB;

CONSEJO: El MSB es el bit situado más a la izquierda y el LSB es el bit situado más a la derecha.

Introducción

Los microcontroladores modernos, como el PIC, son capaces de realizar tareas matemáticas sencillas con números de 8 bits a muy alta velocidad. A medida que los cálculos se hacen más complejos o los números se elevan por encima de un valor de 8 bits, entonces el tiempo de ejecución se alarga drásticamente. Flowcode permite realizar cálculos complejos utilizando números de hasta 16 bits y se encarga de todas las complejidades. Sin embargo, éstas pueden ralentizar la ejecución del programa.

Fondo

- Variables - Ejemplo 1. Añadir entradas digitales - ¿Dónde está el fuego?
- Flowcode Wiki - Creación de variables.
- Entradas digitales - Ejemplo 1. Añadir entradas digitales - ¿Dónde está el fuego?
- Flowcode Wiki - Propiedades de los iconos de cálculo.
- Sección 1 - Introducción a los microcontroladores.

Objetivos

- Crear y utilizar una **variable**.
- Configura un icono de **cálculo** para realizar cálculos aritméticos y lógicos.
- Crear y manipular variables.
- Realiza cálculos.
- Utiliza LED con resistencias limitadoras de corriente.

Tareas

1. Crea un diagrama de flujo que:

- utiliza una variable llamada 'contador'que contiene un valor inicial de '1';
- muestra en los LEDs el valor almacenado en la variable 'contador'.
- Cambia la velocidad de simulación en 'Build > Project Options... > General Options'a 'Normal';
- simule el programa para comprobar que funciona. modifique el Programa 1 por:
- añadiendo un icono de Cálculo para duplicar el valor almacenado en la variable 'contador';
- mostrando este nuevo valor en los LED.
- usando un bucle infinito para repetir estos pasos continuamente con un retardo de 300ms entre ellos. ¿Qué se ve? (Esto se llama "luz de marcha").
- sustituyendo "multiplicar por 2" por "contador = contador + 1". ¿Qué ves ahora? (Acabas de programar un contador binario).

2. Modificar el Programa 1 para que muestre el resultado de los siguientes cálculos en los LEDs del puerto B:

- 45 + 52;
- 45 Y 52;
- 45 O 52;
- NO 45;
- (1+3)*(6/2);
- VAR2 = VAR1 * 3 (donde la variable 'VAR1'almacena el número 18).Sobre el papel, comprueba si los resultados son correctos.

Introducción

Repetición de un conjunto de instrucciones, durante un número exacto de veces, **WHILE** o **UNTIL** a
es una de las operaciones de programación más potentes.
SUGERENCIA: La función de simulación lenta o "Step Over" del simulador Flowcode es útil para depurar programas complejos.

Fondo

- Flowcode Wiki - Propiedades del icono de bucle.
- Flowcode Wiki - Propiedades del icono de punto de conexión.
- Flowcode Wiki - Creación de variables.
- Sección 1 - Introducción a los microcontroladores.
- Sección 2 - Utilización de los bloques electrónicos.
- Sección 4 - Flowcode Primer Proyecto. Añadir salidas digitales - Encender el LED.

Objetivos

- Crea y utiliza un programa de "semáforo en marcha" utilizando el método de "multiplicar por dos".
- Crea y utiliza un programa de "luz de marcha" utilizando el método "shift-right".
- Crear y rellenar un array.
- Crear un **bucle** condicional.

Tareas

1. Escribe un programa para:

- haz un contador binario de 8 bits, usando un icono de Bucle, para contar de 0 a 255, luego reinicia y repite la cuenta; muestra el valor del contador en los LEDs del puerto B.

modifique su programa para

- hacer que el contador cuente de 0 a 255 y luego vuelva a contar hasta '0':

CONSEJO: utilice dos bucles dentro de un bucle infinito para que el proceso se repita indefinidamente;

- Descarga el programa en el microcontrolador y pruébalo a toda velocidad.

2. ¿Recuerdas a KITT de Knight Rider o a los robots Cylon de Battlestar Galactica?

Escribe un programa para hacer una simple 'luz de marcha' que vaya del puerto B, bit 0 al puerto B bit 7 y luego de vuelta al puerto B bit 0, repetidamente:

- Utiliza el método de multiplicar por dos;
- Prueba a utilizar el método "shift right";

Modifica tu programa para crear una luz de marcha de 16 bits, utilizando los LEDs de los puertos A y B. ***CONSEJO: Utiliza sólo bucles, sin decisiones.*** (Descarga el programa en el microcontrolador y pruébalo). Crea un diagrama de flujo que contenga un array de cuatro variables, llamado 'Matriz[x]'que almacene

3. Crea un diagrama de flujo que contenga un array de cuatro variables, llamado 'Matriz[x]'que almacene los siguientes valores: **Matriz[0] =129 Matriz[1] =66 Matriz[2] =36 Matriz[3] =24**(Muestra las salidas en los LEDs del puerto B).

- Utiliza dos bucles 'do-while'para crear una secuencia infinita: Matrix[0]-Matrix[1]-Matrix[2]-Matrix[3]- Matrix[2]-Matrix[1]-Matrix[0]-Matrix[1]- ;
- Refiérete a las cuatro variables como '**Matriz[x]**'donde 'x'es una variable separada, conocida como el índice de la matriz. (Descarga el programa en el microcontrolador y pruébalo).

Introducción

Añadir **entradas** digitales a un circuito de microcontrolador es bastante fácil, pero supone un gran paso adelante. Esto permite que señales externas influyan en cómo reacciona el programa.

Fondo

- Sección 1 - Introducción a los microcontroladores.
- Sección 2 - Utilización de los bloques electrónicos.
- Sección 4 - Flowcode Primer Proyecto. Añadir salidas digitales - Encender el LED.

Objetivos

- Datos de entrada de los **interruptores**.
- Utiliza bucles para crear secuencias de LED.
- Configurar un **icono de entrada**.

Tareas

1. Escribe un programa para mostrar el estado de los interruptores conectados a un puerto elegido, en los LEDs conectados a un puerto diferente. ej. cuando se pulsa un
Modifique el programa para que
- el LED permanezca encendido durante 2s.
- al pulsar el interruptor '0'se encienda el LED 1.
- al pulsar el interruptor '1' se encienda el LED 2 y así sucesivamente.
- cuando se pulsa el interruptor '7', no ocurre nada.
Explora tantas combinaciones como puedas.
(Descarga los programas en el microcontrolador y pruébalos).

2. Escribe un programa para crear un contador que:
- contenga dos bucles.
- cuente hacia arriba cuando se pulse el interruptor '0'.
- cuente hacia abajo cuando se pulse el interruptor '1'.
- muestre la cuenta en la matriz de LED de un puerto adecuado.
(Descarga los programas en el microcontrolador y pruébalos).

3. Escribe un programa 'luz de marcha' que
- contenga dos bucles.
- haga girar los LEDs a la izquierda cuando se pulse el interruptor '0'.
- haga que los LEDs funcionen a la derecha cuando se pulse el interruptor '1'.
- muestra el recuento en la matriz de LED de un puerto adecuado.
(Descarga los programas en el microcontrolador y pruébalos).

Introducción

Los programas anteriores incluían una toma de decisiones sencilla, mediante bucles y puntos de conexión.

Ahora veremos en detalle el icono de **Decisión**, ampliamente conocido como la estructura 'if...then...else', probablemente la línea de comandos más utilizada en cualquier programa.

Fondo

- Flowcode Wiki - Propiedades de los iconos de decisión.
- Flowcode Wiki - Propiedades del icono de punto de conexión.
- Sección 1 - Introducción a los microcontroladores.
- Sección 2 - Utilización de los bloques electrónicos.

Objetivos

- Configurar iconos de **decisión** y, por tanto, añadir bifurcaciones condicionales a un programa.
- Controla la **frecuencia** con la que parpadean los LED.
- Utiliza LEDs para mostrar los niveles lógicos de salida.
- Utilizar memoria temporal.

Tareas

1. Escribe un programa que utilice interruptores para producir una secuencia inversa en los LEDs:

- cuando se pulsa el interruptor '0'se ilumina 'LB7';
- cuando se pulsa el interruptor '1', se

enciende 'LB6'; y así sucesivamente...

2. Escribe un programa que cree un contador de 8 bits, contando de '0'a '255'y luego de vuelta a '0'repetidamente:

- utilizando iconos de Decisión en lugar de iconos de Bucle.
- mediante dos conmutadores conectados al puerto B, bits 0 y 1;
- cuenta atrás cuando se pulsa el interruptor '0';
- cuenta atrás cuando se pulsa el interruptor '1';
- muestra el recuento actual en los LEDs conectados al puerto A;
- guarda este programa, descárgalo en el microcontrolador y pruébalo.

3. Escribe un programa que cuente desde '0'hasta un valor almacenado en una variable llamada 'count' cuando se pulsa el interruptor '0'y luego espere hasta que se pulse el interruptor '1'antes de contar hacia atrás hasta '0':

- mediante dos conmutadores conectados al puerto B, bits 0 y 1;
- mostrando el valor actual de la cuenta en los LEDs del puerto

A; guarda este programa, descárgalo en el microcontrolador y

pruébalo.

4. Escribe un programa que haga que los ocho LEDs del puerto B parpadeen encendiéndose y apagándose a una frecuencia de 1Hz, es decir, tardando un segundo en realizar un ciclo de encendido y apagado. Además:

- los LED parpadean más rápido si se pulsa el interruptor '0';
- parpadean más lentamente si se pulsa el interruptor '1';

guarda este programa, descárgalo en el microcontrolador y pruébalo.

Tareas

5. Escribe un programa que haga que los ocho LEDs del puerto B se enciendan cuando se pulse el interruptor '0'la primera vez y se apaguen cuando se vuelva a pulsar:

Guarda este programa, descárgalo en el microcontrolador y pruébalo.

6. Un coche tiene dos luces interiores, una en la parte delantera y otra en la trasera.

Escribe un programa para simular este escenario utilizando LEDs y cinco interruptores para controlarlos.

- Los interruptores '0'a '3'representan interruptores de puerta que indican si una puerta está abierta o no;
- El interruptor'4'indica si el maletero está abierto o no.
- Enciende ambos LEDs cuando se abre cualquier puerta;
- Enciende sólo el LED 'trasero' cuando se abre el maletero;

Guarda este programa, descárgalo en el microcontrolador y pruébalo.

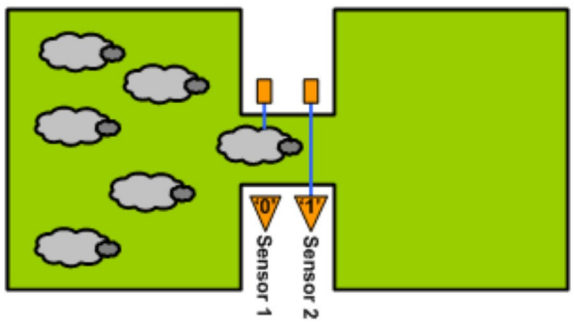
CONSEJO: suponga que los interruptores están cerrados cuando las puertas están abiertas. Esto puede ser más fácil de simular con interruptores "pulsar para hacer".

7. El volante de un coche tiene interruptores que controlan las luces exteriores. Escribe un programa para simular el control de las luces.

- Utilice un interruptor para controlar el indicador de dirección izquierdo (elija un LED relevante), que parpadea en
- durante 250 ms y luego se apaga durante 250 ms repetidamente hasta que se suelta el interruptor.
- Utilice otro interruptor para controlar el indicador de dirección derecho (elija un LED relevante), de la misma manera.
- Utiliza dos LED como luces de freno controladas por un interruptor que se encienden mientras está pulsado.
- Crea faros que se enciendan al pulsar un interruptor y permanezcan encendidos hasta que se vuelva a pulsar.
- Termina con un par de faros antiniebla de la misma manera.

CONSEJO: No intentes escribir este programa de una sola vez. Divídelo en subsecciones y resuelve cada una por separado antes de ponerlas todas juntas.
Para hacerlo más fácil, utilice la función de etiquetado de Flowcode para etiquetar los interruptores y los LED.

8. Seis ovejas pueden deambular entre dos campos. Hay dos sensores entre los campos. Escribe un programa que cuente y muestre el número de ovejas en cada campo. Simule este escenario utilizando dos interruptores para representar los sensores.
Muestra los resultados en forma binaria en la matriz de LED (utiliza cuatro LED para el campo oeste y cuatro para el campo este).
Utiliza dos interruptores para representar los sensores.



CONSEJO: Suponga que cada oveja es más larga que la distancia entre los sensores. Piense en las distintas situaciones que podrían producirse. Una oveja puede activar un sensor y luego retroceder. ¿Puede una oveja activar ambos sensores y luego retroceder? ¿Cuándo se considera que una oveja está en el campo este?

Introducción

El uso de LED para mostrar las salidas puede ser limitante.

La pantalla LCD es una forma alternativa de mostrar datos, tanto letras como números, para los humanos "no binarios".

Fondo

- Sección 1 - Introducción a los microcontroladores.
- Sección 2 - Utilización de los bloques electrónicos.
- Ejemplo 3. La pantalla LCD - Mensajes

Objetivos

Crear, rellenar y manipular **variables de cadena**.
Controla la visualización de texto y números en una **pantalla LCD**.
Utilizar un LCD como dispositivo de salida para el microcontrolador. Configure una **macro Component** para la **pantalla LCD**.

Tareas

1. Escribe un programa que muestre el texto "Hola Mundo" en el centro de la línea inferior de la **pantalla LCD**.

2. Escriba un programa que muestre una cuenta creciente (decimal) en la pantalla LCD. Modifica el programa para que cuente hacia arriba cuando un interruptor es presionado y cuente hacia abajo cuando un interruptor diferente es presionado (usa **Bucles** o **Decisiones**).

3. Escriba un programa para mostrar el estado de los interruptores conectados al primer puerto. Cada vez que se pulsa un interruptor, el LED correspondiente del segundo puerto se ilumina y el valor del equivalente **decimal** se muestra en la pantalla LCD.

4. Escriba un programa para mostrar el estado de los interruptores conectados al primer puerto en los LED del segundo puerto y en la línea superior de la pantalla LCD y, a continuación:

- multiplica este número **binario** por 100.
- mostrar el resultado en la línea inferior de la pantalla LCD, con "[x 100 =]" delante.

Tareas

5. Escribe un programa que desplace las líneas de texto dadas a continuación, una línea cada vez. Inicialmente, el texto se centra en la línea inferior de la pantalla durante 2s. Luego se desplaza hacia arriba para centrarse en la línea superior durante 2s, para ser reemplazado en la línea inferior por la siguiente línea de texto, y así sucesivamente.

Texto:

"Sólo hay" "10

tipos"

"de la gente" "Los

que" "entienden"

"BINARIO"

"y los que" "NO".

(Encierre el programa en un **bucle infinito** y pruébelo en la **pantalla LCD**).

Introducción

En muchos dispositivos electrónicos se utiliza un **teclado numérico**, y en algunos (por ejemplo, en los teléfonos móviles), es utilizado como teclado numérico y también como forma de escribir texto en lugar de números. Hay doce botones en el teclado, pero éste está conectado al microcontrolador por sólo ocho líneas. Este problema se resuelve utilizando **multiplexación**.

Fondo

- LCD - Ejercicio 8 - Programación de LCDs.
- Flowcode Wiki - Funciones de manipulación de cadenas.
- Sección 1 - Introducción a los microcontroladores.
- Sección 2 - Utilización de los bloques electrónicos.

Objetivos

- Introduzca texto y números desde un teclado y visualice mensajes en la pantalla LCD.
- Utilice **el código ASCII** para transmitir estos datos.
- Utiliza **entradas multiplexadas**.
- Configurar una **macro Componente** para el **teclado**.

Tareas

1. Muestra en la **pantalla LCD** los números que se pulsan en el **teclado**.
   - Muestra un numero de uno en uno mientras el boton del teclado este presionado.
- ¿Puede reescribir este programa sin utilizar la **macro** Keypad **Component**?
- Amplíe este programa para mostrar los números que se pulsan en el teclado uno tras otro en la fila superior de la pantalla LCD.

A ver si puedes perfeccionar el programa para:

- Borra la pantalla cuando se pulsa '#'.
- Muestra un máximo de quince caracteres y muestra una advertencia en la fila inferior de la pantalla LCD cuando se supera este máximo.

2. Escribe un programa para:

- suma dos números, inferiores a 9999, introducidos mediante el teclado;
- mostrar los dos números, el '+'y el '='y la suma resultante en la fila superior de la pantalla LCD;
- mostrar un aviso en la fila inferior cuando se supere el límite de '9999';

3. Escribe un programa para un simple juego de adivinanzas, donde:

- un jugador tiene que adivinar un número entre '0'y '9';
- el número secreto está preprogramado en el PIC;
- la pantalla LCD muestra, en la fila superior, el último cálculo introducido mediante el teclado;
- la pantalla LCD muestra un mensaje, en la fila inferior, indicando si la estimación es demasiado alta o demasiado baja;

Amplía el programa 3 para que: el número secreto esté en el rango de '0'a '255'. Amplía el programa 3 de nuevo para que: el número secreto esté en el rango de '0'a '9999'.

4. Escribe un programa para utilizar el **teclado**, como en un teléfono móvil, para introducir texto en el microcontrolador.

- Utiliza **el código ASCII** para transmitir los datos.
- Utilice el carácter '*'para un espacio.
- Borra la pantalla cuando se pulsa '#'.
- Mostrar un mensaje en la fila inferior cuando el texto tiene más de diez caracteres.

Introducción

La MCU 16F18877 PIC acepta 35 entradas analógicas independientes. Los más nuevos pueden tener incluso más. Una señal **analógica** en una de estas entradas se puede traducir en un
Número binario digital de 10 bits. Podemos elegir utilizar sólo los ocho bits más significativos de este número de 10 bits o utilizar el número completo de 10 bits. Ten en cuenta que trabajar con números de 10 bits en un microcontrolador de 8 bits como el PIC MCU, requiere una cuidadosa escritura del programa.

Fondo

- LCD - Ejercicio 8 - Programación de LCDs.
- Flowcode Wiki - Funciones de manipulación de cadenas.
- Sección 1 - Introducción a los microcontroladores.
- Sección 2 - Utilización de los bloques electrónicos.

Objetivos

- Crear registradores de datos, utilizando datos de 8 y 10 bits del ADC.
- Configura una entrada analógica.
- Introducir datos mediante interruptores.
- Introduce la información de los sensores de luz y temperatura.
- Configurar y utilizar la **EEPROM**.
- Desplazarse por los datos de **la EEPROM**.
- Muestra texto y datos numéricos en la pantalla LCD.

Tareas

1. Escribe un programa que muestre un número de 8 bits, equivalente a la entrada **analógica** voltaje del sensor de luz en la placa del sensor. Intente conectar un voltímetro a

medir la tensión de entrada **analógica**. (Guarde los siguientes programas y descárguelos en el microcontrolador para probarlos).

2. Modifica el programa de la Tarea 1 para mostrar los datos del '**pote'** de la placa del Sensor. Intenta convertir la salida de 8 bits del ADC en una lectura de tensión entre 0 y 5V, haciéndolo tan preciso como permita el modo de 8 bits. Utilice un **voltímetro** para medir la tensión de entrada analógica.

3. Modifique el programa para que muestre, en la pantalla LCD, un número de 10 bits equivalente a la tensión de entrada **analógica** del 'pote' de la placa del sensor. Utilice un voltímetro para medir la tensión de entrada analógica. Intente convertir la salida de 10 bits del ADC en una lectura de tensión entre 0 y 5V, haciéndolo tan preciso como permita el modo de 10 bits. Utilice un **voltímetro** para medir la tensión de entrada **analógica**.

4. Escribe un programa para controlar la iluminación de una habitación durante un periodo de 24 horas:

- mediante la señal analógica del sensor de luz de la tarjeta Sensor
- almacenar las mediciones de luz en la **EEPROM**.
- muestreo a la mayor velocidad posible, dado que el PIC MCU tiene 256 bytes de memoria **EEPROM** en la placa.
- y mostrando cada muestra con su número de muestra, en la pantalla LCD.
- avanzando por las muestras pulsando el interruptor "0" o retrocediendo pulsando el interruptor "1".

CONSEJO: Aumente la frecuencia de muestreo para no tener que dedicar 24 horas a las pruebas.

Introducción

En los lenguajes de programación basados en código, como "C" y "BASIC", una macro de software se llamaría "subrutina", "función" o "procedimiento". A medida que los programas se hacen más grandes, utilizan ciertas

combinaciones de instrucciones una y otra vez. Estos programas resultan más difíciles de entender y leer. Las rutinas que se reutilizan pueden incluirse en una macro de software, que puede invocarse siempre que sea necesario en el programa principal. El uso de estas macros aligera el programa principal y facilita su lectura.

Fondo

- Flowcode Wiki - Propiedades de **los** iconos de **macros de software**
- Sección 2 - Utilización de los bloques electrónicos

Objetivos

- Utilice macros de software para simplificar la estructura de un programa.
- Crear **macros de software**.
- Utilice el control de bucle cerrado.
- Utiliza PWM para controlar el brillo de los LED.

Tareas

1. Escribe un programa que seleccione y ejecute uno de tres programas diferentes utilizando dos interruptores:

- El interruptor '0'selecciona uno de los tres programas (que has desarrollado anteriormente);
    - 'X': un contador binario ascendente de 8 bits, que se muestra en los LED.
    - 'Y': un contador descendente binario de 8 bits, visualizado en los LEDs.
    - 'Z': una 'luz de marcha' bidireccional de 8 bits, que se muestra en los LED.
- la pantalla LCD muestra un mensaje de texto que identifica el programa seleccionado;
- El interruptor '1'activa el programa elegido al pulsarlo.
- los tres programas se colocan en macros de

software. descarga este programa en el

microcontrolador y pruébalo.

modificar el programa 1 para que:

- Si se pulsa el interruptor '0'mientras se está ejecutando uno de los tres programas, la ejecución se detiene inmediatamente y el foco vuelve al bucle principal y espera una nueva selección.

descarga este programa en el microcontrolador y pruébalo.

modifica de nuevo el programa 1 para que:

- si se pulsa el interruptor '0'mientras se está ejecutando uno de los tres software, la ejecución se detiene y vuelve al bucle principal, como antes, pero almacena el valor mostrado en los LEDs;
- cuando se realiza la siguiente selección, esa macro inicia los LED desde donde lo dejó la anterior, haciendo que la transición entre ellas sea más suave".

descarga este programa en el microcontrolador y pruébalo.

Introducción

En ejercicios anteriores, el microcontrolador no reaccionaba necesariamente a las entradas de forma inmediata porque estaba ocupado haciendo otra cosa. Las funciones de interrupción externa del PIC resuelven este problema. En un 16F18877, las interrupciones externas están en el pin 'RA0-RB7'y en todos los pines de los Puertos A, B, D y E como una 'interrupción por cambio (IOC)'. Si estas interrupciones se inicializan correctamente, entonces un cambio en los pines seleccionados puede hacer que el programa detenga la ejecución inmediatamente y pase a ejecutar la macro de interrupción apropiada. Entonces tenemos lo que se llama una ejecución en 'tiempo real'.

Fondo

- Sección 2 - Utilización de los bloques electrónicos.
- Wiki de Flowcode.

Objetivos

- Crear y utilizar **interrupciones de un solo pin**.
- Crear y utilizar **interrupciones por cambio (IOC)**.
- Utilizar el funcionamiento en tiempo real de un microcontrolador.

Tareas

1. Escriba un programa para cronometrar cuántos segundos han pasado desde que se reinició un programa y muestre el resultado en un LCD. Use una variable llamada **conteo** cuyo valor se muestra en los LEDs (no use una interrupción). Usa un retardo de 1s. Un flanco ascendente en el pin RB0 debe llamar a una macro que sume uno al **conteo**.

Rediseña este programa utilizando una **interrupción** (de un solo pin) en RB0. Ahora rediseñalo usando ambos tipos de

interrupción externa para que:

- la activación de la interrupción de un solo pin incrementa '**count**'(count = count + 1)
- la activación de la interrupción IOC disminuye '**count**'(count = count - 1)

2. Escribe un programa para hacer un dado electrónico que :

- cuenta de 1 a 12;
- visualizar el resultado en la pantalla LCD;
- empieza a 'rodar' cuando se pulsa el interruptor '0';
- deja de 'rodar'cuando se vuelve a pulsar el interruptor '0'. descarga este programa al microcontrolador y pruébalo.

CONSEJO: La pantalla LCD debe mostrar los números del 1 al 12, uno tras otro, una y otra vez rápidamente, a intervalos de 20 ms - demasiado rápido para verlo con el ojo humano.

modificar el programa 2 para que:

- el dado sigue 'rodando'mientras se mantenga pulsado el interruptor '0';
- deja de 'rodar' al soltar el interruptor;
- en ese momento muestra el número en la pantalla LCD.

Tareas

3. Escribe un programa para hacer un temporizador de reacción que :

- enciende todos los LEDs inicialmente;
- los mantiene encendidos durante unos 6 s;
- los apaga y pone en marcha un temporizador;
- detiene el temporizador cuando el jugador pulsa el interruptor '0';
- y muestra el "tiempo de reacción" resultante en la

pantalla LCD. (Utilice una variable que se incremente cada 10ms).

descarga este programa en el microcontrolador y pruébalo.

modificar el programa 3 para limitar el tiempo permitido al tamaño de la variable utilizada y:

- aparece un mensaje en la pantalla LCD cuando se supera este tamaño;
- incluye una trampa para evitar trampas simplemente manteniendo pulsado el

interruptor '0'continuamente. descarga este programa al microcontrolador y pruébalo.

Introducción

El otro tipo de función de interrupción en Flowcode es la interrupción del temporizador. Estos le permiten realizar tareas de software en intervalos de tiempo predeterminados con precisión - una característica realmente útil cuando se desarrollan aplicaciones de tiempo crítico y relojes.

Fondo

- Flowcode Wiki - ¿Qué es una pantalla de 7 segmentos?
- Sección 2 - Utilización de los bloques electrónicos.

Objetivos

- Crear y utilizar la interrupción del temporizador.
- Utiliza el preescalador para crear intervalos de tiempo precisos.
- Dispara el temporizador utilizando el cristal o un evento externo.

Aritmética del temporizador

El 16F18877 tiene varios temporizadores, pero sólo veremos dos: 'TMR0'(Timer 0) y 'TMR1'(Timer 0).

TMR0 puede ser disparado por el cristal o por una transición en el pin 'T0CKI' que es mapeable en los puertos

B o C. El reloj interno tiene una frecuencia de 'frecuencia de reloj de cristal'/4, es decir, 32MHz/4 = 8MHz.

El preescalador TMR0 puede ajustarse de 1:2 a 1:32768. Para este ejercicio, ajústalo a 1:2, de forma que cada 8MHz/65536/2 pulsos de reloj hagan que el TMR0 aumente en 1. Esto ocurre a una frecuencia de 8MHz/65536/2 = 61.035 Hz.
El Postescalador se deja en 1:1 o la frecuencia se dividirá aún más.
Cada vez que este temporizador de 16 bits se 'desborda' (alcanza 65536), genera una interrupción. Esto ocurre con una frecuencia de 8MHz/65536/2= 61.035 Hz, de modo que el programa principal se detiene 61.035 veces por segundo y por tanto la macro de interrupción del temporizador se ejecuta 61.035 veces por segundo.
En lugar de utilizar el cristal, este temporizador también puede ser "cronometrado" por un evento externo, como cuando se mide la velocidad del motor, etc.

TMR1 puede ser disparado por el oscilador de cristal o por una transición en el pin 'T1CKI'todos los pines del puerto A y puerto C '.

Tareas

1. Escriba un programa para producir un temporizador preciso de 'segundos' que muestre el resultado en la pantalla LCD y se inicie cuando se reinicie el microcontrolador. Usa un retardo de 1s. No uses una interrupción del temporizador.

(Descarga este programa en el microcontrolador y pruébalo utilizando tu reloj).
Reescribe el programa utilizando una interrupción del temporizador.
2. Escriba un programa para crear un temporizador de baloncesto que se inicie al pulsar el interruptor 0 y muestre el tiempo transcurrido en la pantalla LCD. Haz que los LEDs se enciendan y apaguen cuando hayan transcurrido 30s (el tiempo permitido para que el equipo con el balón haga un intento de gol).
CONSEJO: Utilice una interrupción de un solo bit en el pin RB0 para iniciar la temporización).

Tareas

3. Escribe un programa para producir un reloj preciso que muestre el tiempo transcurrido desde la última puesta a cero, en horas, minutos y segundos en la pantalla LCD (prueba con un reloj).

Modifica este programa para que:

- El interruptor '0'detiene el reloj cuando se pulsa por primera vez.
- Los interruptores '1', '2'y '3'permiten cambiar la hora visualizada por la hora real.
- El interruptor '0'reinicia el reloj cuando se pulsa por segunda vez.

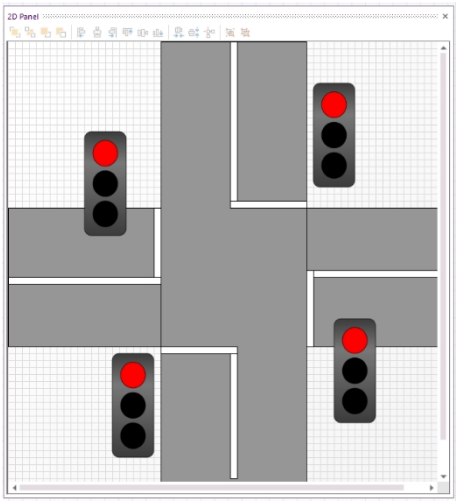
4. Escribe un programa para producir un temporizador que cuente desde 01:00:00 hasta 00:00:00 en segundos y luego encienda todos los LEDs.

(Descárgalo en el microcontrolador y pruébalo con tu reloj).

Flowcode dispone de un sofisticado motor de simulación. Esto le permite establecer un gran número de retos que puede completar y aprender haciéndolos. En las páginas siguientes te damos algunas sugerencias de trabajos adicionales que podrías realizar.

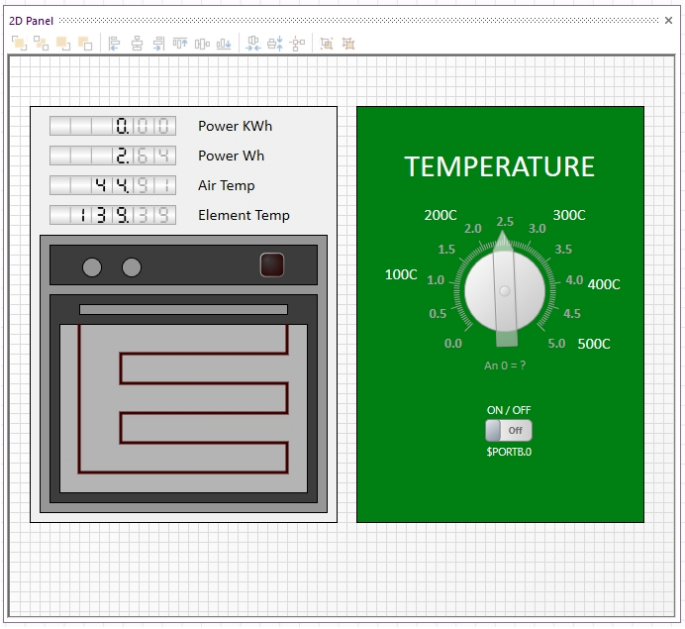
Semáforos

- Crea un cruce de tráfico. Para ello utilice el componente SEMÁFORO que encontrará en SISTEMA para colocar 4 semáforos en un panel.
- Puede utilizar formas básicas en CREACIÓN para crear una representación visual de la unión.
- Crea un programa que gestione la secuencia de semáforos: rojo...ámbar...verde para gestionar el tráfico en el cruce.



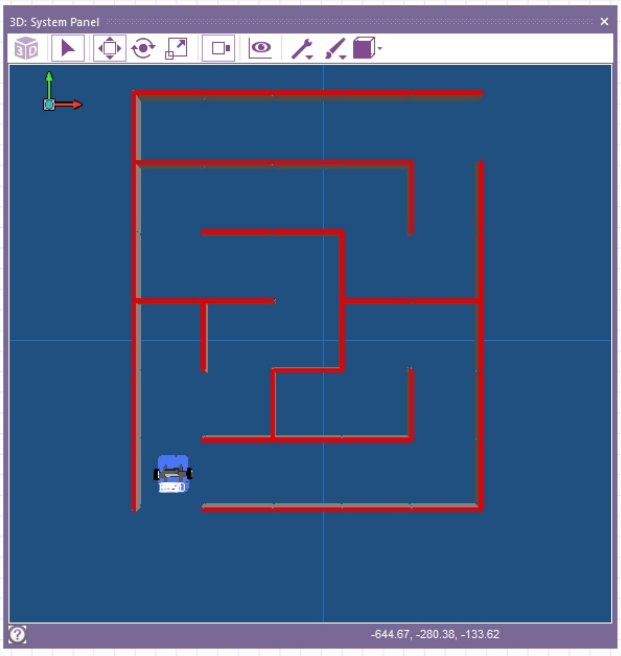
Horno

- Utilizando el componente Horno (en Sistema) cree un panel 2D con un interruptor On Off y un potenciómetro que le permita ajustar la temperatura objetivo en C.
- Escribe un programa que permita al usuario encender y apagar el horno y ajustar la temperatura.
- Modifica tu programa para incluir un gráfico que te muestre la temperatura a lo largo del tiempo.
- AVANZADO: investigar el uso de componentes DSP para controlar el horno utilizando un sistema de flujo de datos.



Allcode robot buggy

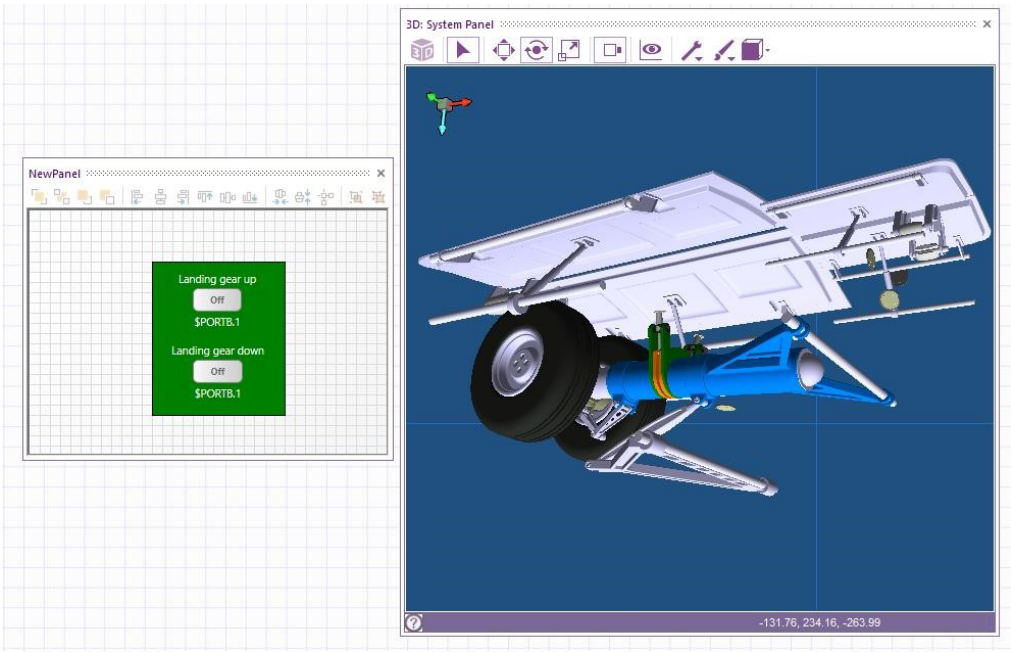
- Inicie un nuevo programa. Abra un panel 3D. Desde SYSTEM seleccione MAZE GENERATOR con una anchura por defecto de 5 y una longitud de 6.



- Desde HARDWARE selecciona el buggy FORMULA ALLCODE y añade uno al panel 3D.
- Ya tienes un laberinto robótico virtual completo. En la Wiki podrás averiguar cómo funcionan los sensores del robot.
- Desarrolla un programa para resolver el laberinto. (Pista: intenta seguir la pared de la izquierda).

Tren de aterrizaje de aviones

- Configure un nuevo panel 3D. Seleccione SISTEMAS...PALANCA DE ATERRIZAJE. En el panel 3D aparecerá un tren de aterrizaje típico de un avión.
- Configura un nuevo panel 2D. Coloca un interruptor separado para Arriba y Abajo en el panel.
- Desarrolla un programa que utilice los interruptores para controlar el tren de aterrizaje. Tendrás que detectar si el tren de aterrizaje está completamente arriba o abajo como parte de tu programa.



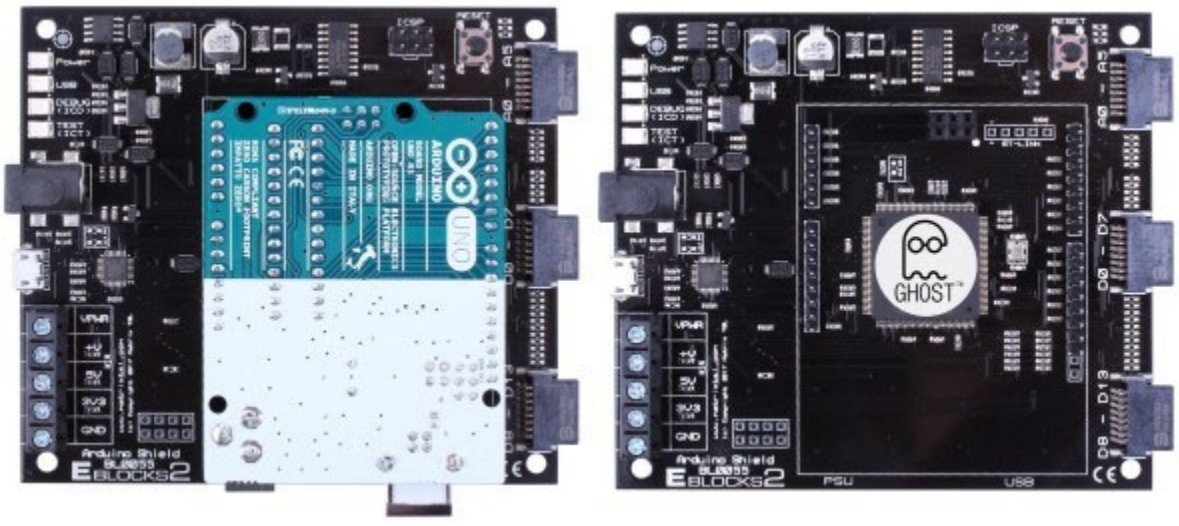
Anexo 1:

Arduino ajustes

ARDUINO: SECCIÓN A

BL0055 Escudo Arduino

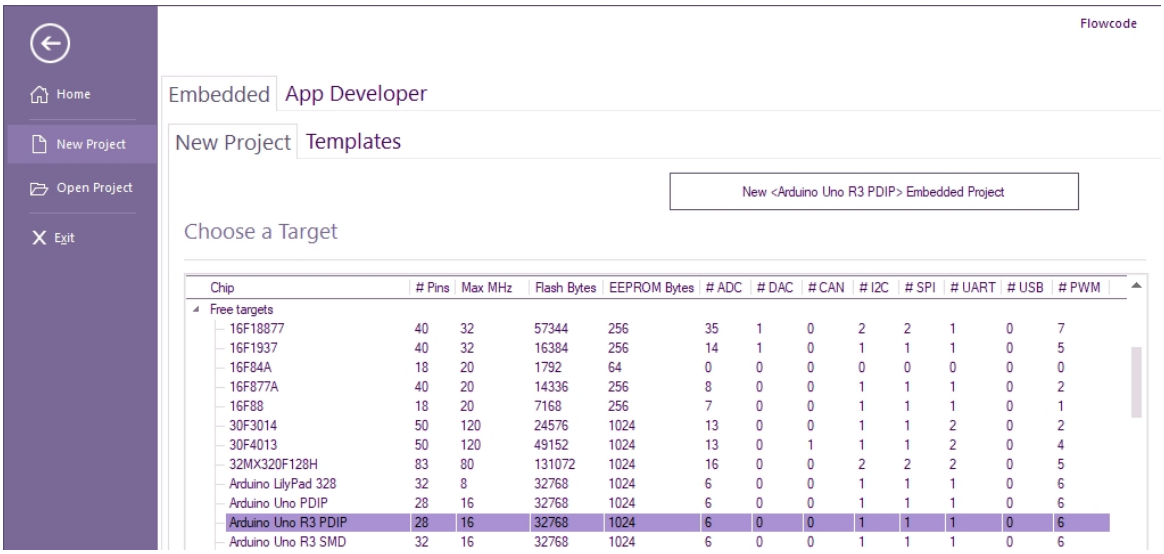
- La placa tiene tres puertos, denominados **A0-A5**, **D0-D7** y **D8-D13**.
- Los puertos **D0-D7** ofrecen una funcionalidad completa de 8 bits.
- Los puertos **A0-A5** y **D8-D13** tienen una funcionalidad de 6 bits.
- Se puede alimentar desde una fuente de alimentación externa de 7,5 V a 9 V o desde una fuente USB.
- Si se pulsa el interruptor Reset, se reiniciará el programa almacenado en el Arduino.
- La placa es programable por USB mediante un chip de programación. Esto se encarga de la comunicación
- entre **Flowcode** y el dispositivo **Arduino**.
- El Arduino ejecuta una instrucción por cada pulso de reloj que recibe.
- (Nota - una sola instrucción NO es lo mismo que un solo símbolo Flowcode, que se compila en C y luego en ensamblador y probablemente da como resultado una serie de instrucciones).
- Este dispositivo utiliza un **cristal de 16 MHz**.
- La placa detectará si se debe utilizar una fuente de alimentación externa o una fuente de alimentación USB.
- Uso de la herramienta AVR ISP de Microchip a través de la cabecera ICSP.
- Normalmente se suministra con un dispositivo Arduino Uno.
- Proporciona alimentación a las tarjetas E-blocks posteriores a través de los conectores de puerto.
- Contiene el chip Matrix Ghost que permite la depuración en circuito en tiempo real y la monitorización de pines cuando se combina con Flowcode.



ARDUINO: SECCIÓN B

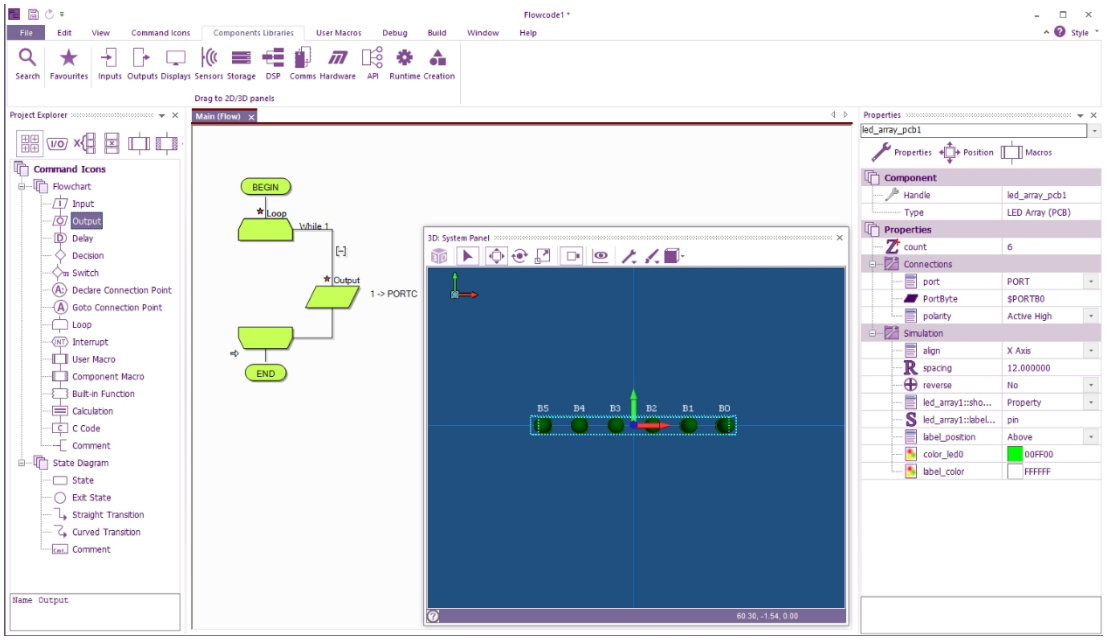
Selección de Arduino en Flowcode

Al abrir Flowcode, se le presentará la pantalla de 'Bienvenida'. Haga clic en **Nuevo Proyecto**.



Chip	# Pins	Max MHz	Flash Bytes	EEPROM Bytes	# ADC	# DAC	# CAN	# I2C	# SPI	# UART	# USB	# PWM
Free targets												
16F18877	40	32	57344	256	35	1	0	2	2	1	0	7
16F1937	40	32	16384	256	14	1	0	1	1	1	0	5
16F84A	18	20	1792	64	0	0	0	0	0	0	0	0
16F877A	40	20	14336	256	8	0	0	1	1	1	0	2
16F88	18	20	7168	256	7	0	0	1	1	1	0	1
30F3014	50	120	24576	1024	13	0	0	1	1	2	0	2
30F4013	50	120	49152	1024	13	0	1	1	1	2	0	4
32MX320F128H	83	80	131072	1024	16	0	0	2	2	2	0	5
Arduino LilyPad 328	32	8	32768	1024	6	0	0	1	1	1	0	6
Arduino Uno PDIP	28	16	32768	1024	6	0	0	1	1	1	0	6
Arduino Uno R3 PDIP	28	16	32768	1024	6	0	0	1	1	1	0	6
Arduino Uno R3 SMD	32	16	32768	1024	6	0	0	1	1	1	0	6

Selecciona **Arduino Uno R3 PDIP** de la lista de objetivos libres. Haz clic en el botón **"Nuevo <Arduino...** Aparecerá el entorno estándar de Flowcode. Un diagrama de flujo puede ahora ser desarrollado en un programa que puede ser probado en el modo de simulación Flowcode, o guardado y compilado en la placa Arduino.



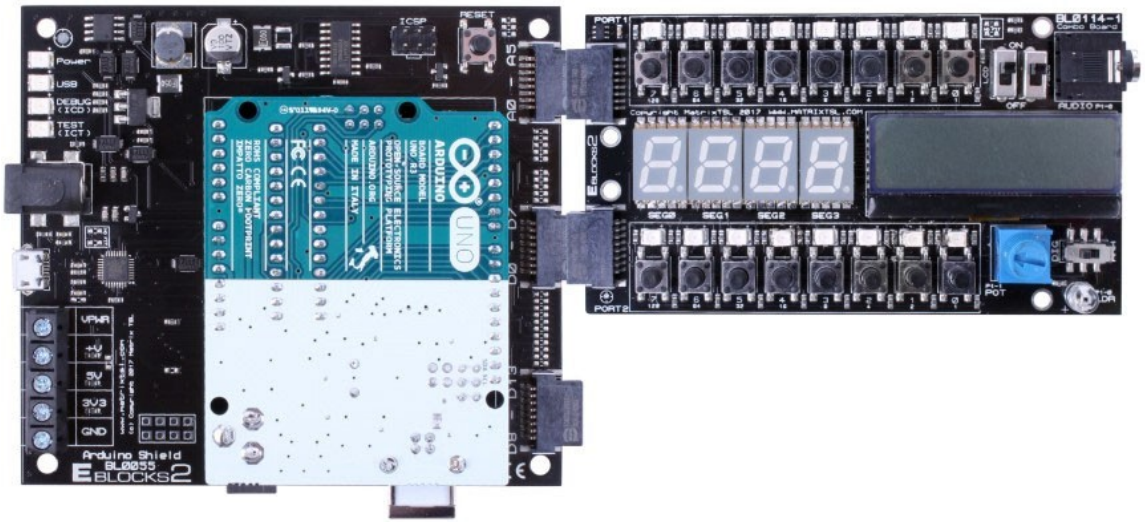
Siga los Ejemplos y Ejercicios, teniendo en cuenta los cambios en los puertos cuando sea necesario. Por ejemplo, arriba se muestra cómo se vería **el primer programa Flowcode** (página 42) para un usuario de Arduino. Aquí, los usuarios de Arduino están utilizando **PORTC** en lugar de **PORTA**.

(PORTC en el Arduino 'Maps'to PORTA de la placa Combo)

ARDUINO: SECCIÓN DC

E-blocks2:

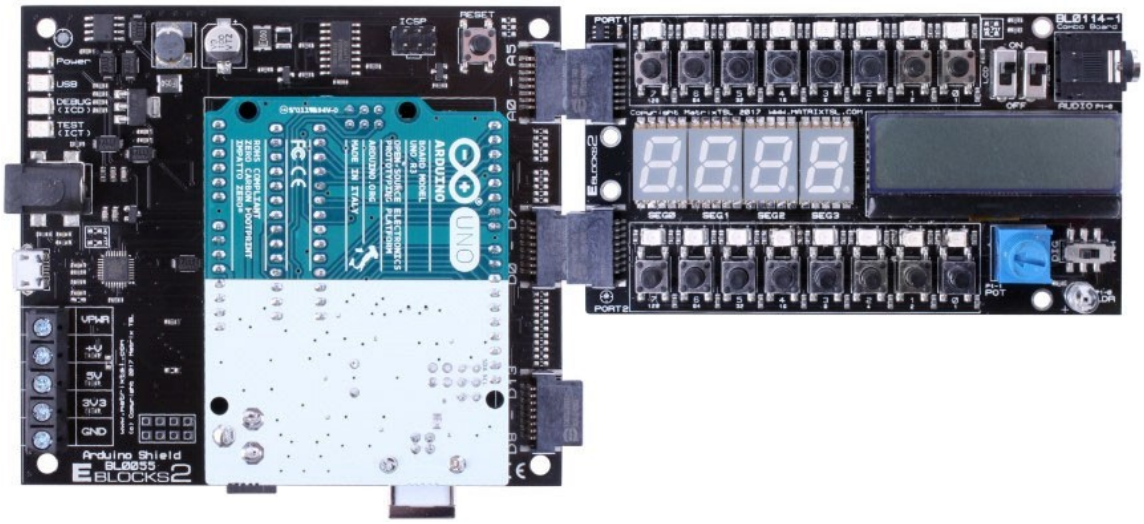
Eblocks2 utiliza las placas 'Click'para sus conexiones SPI. Usando la placa 'Click'E-block BL0106, puedes poner la placa en el puerto (D8-D13) como se muestra en la imagen de abajo:



ARDUINO: SECCIÓN D

Configuración del hardware:

Este diagrama muestra cómo configurar el hardware E-blocks con Arduino. Conecta tu Arduino a la placa BL0055 como se muestra, y luego la placa combo a los puertos etiquetados **(A0-A5)** y **(D0-D7)**.



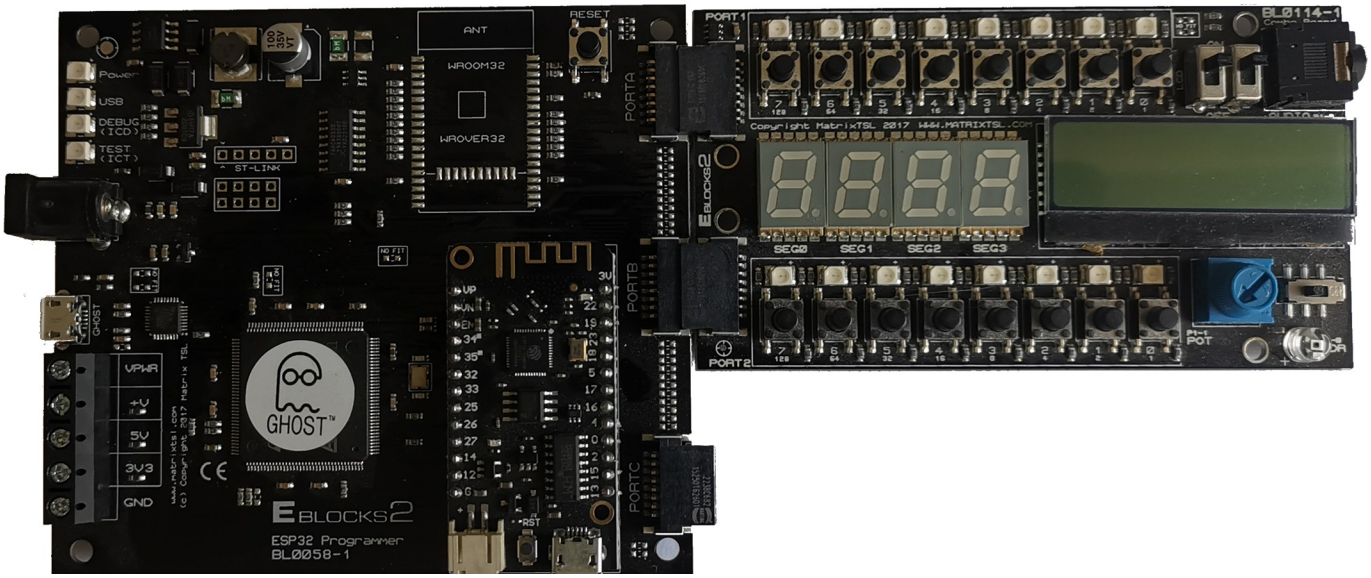
Nota: A pesar de tener dos conexiones de puerto de hardware entre la placa de desarrollo EB0114 y el BL0055 Shield, el Arduino Uno sólo puede proporcionar 6 conexiones de E/S de propósito general en el puerto C, (A0-A5). Por lo tanto, los LEDs '6'y '7'y los interruptores '6'y '7'en el puerto 1 de la placa de desarrollo, no se pueden utilizar con el Arduino Uno.

Con el fin de programar la placa Arduino Uno directamente desde Flowcode, debe asegurarse de que los controladores adecuados están instalados. Le recomendamos que visite el sitio de Arduino y descargue los controladores más recientes desde allí.

Anexo 2:

ESP32 ajustes

BL0058: SECCIÓN A



Nota: A pesar de tener dos conexiones de puerto de hardware entre la placa de desarrollo EB0114 y el BL0055, el LOLIN32 sólo puede proporcionar 3 conexiones de E/S de propósito general en el puerto C, (IO16, IO17.e IO23). Por lo tanto, evite conectar la placa de desarrollo BL0114 al puerto C .
Cualquier pin que sea 34 o superior es sólo de entrada.

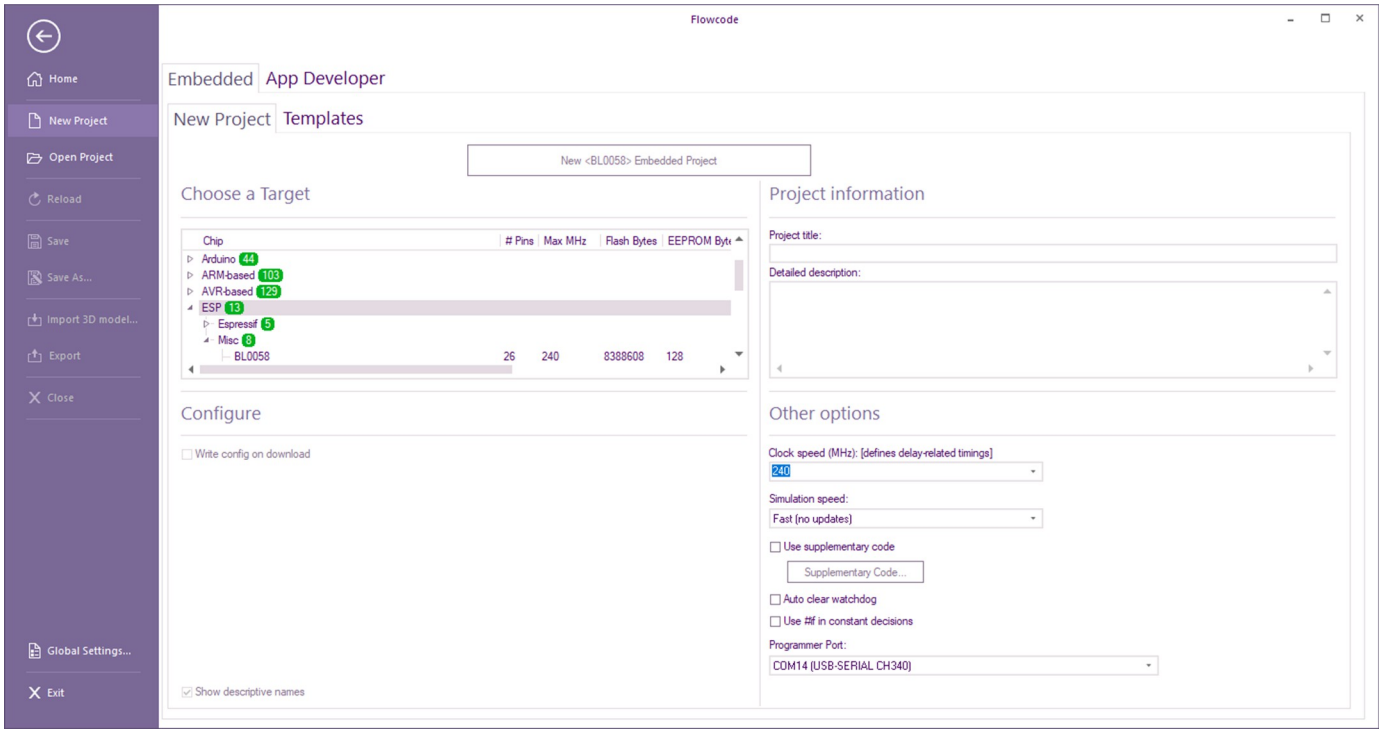
Para programar el BL0058(LOLIN32) directamente desde Flowcode, debe asegurarse de que los drivers apropiados están instalados. Le recomendamos que visite https://www.wch-ic.com/downloads/ CH341SER_ZIP.html y descargue los últimos drivers desde allí.

Usted tendrá que instalar el toolchain ESP32.
Las instrucciones se encuentran en: https://www.flowcode.co.uk/wiki/index.php?title=Cadenas_de_herramientas_de_compilación#ESP32_Toolchain
Para programar el ESP32 Eblock, conecte el microconector USB directamente al ESP32 LOLIN32 y no al BL0058 ya que eso es sólo para depuración Ghost.

BL0058: SECCIÓN B

Selección de BL0058 en Flowcode

Con el USB conectado al ESP32 LOLIN abre Flowcode
Al abrir Flowcode, se le presentará la pantalla de 'Bienvenida'. Haga clic en **Nuevo Proyecto**.
Seleccione la flecha **ESP** y luego **Misc**.
Seleccione **BL0058,** luego seleccione el puerto com correcto en la parte inferior de **Otras opciones**
haga clic en el botón **Nuevo <BL0058 >Proyecto integrado** de arriba.



Aparecerá el entorno estándar Flowcode. Ahora se puede desarrollar un diagrama de flujo en un programa que se puede probar en el modo de simulación Flowcode, o guardar y compilar en la placa ESP32 LOLIN.

La siguiente tabla muestra cómo los pines de la ESP32 LOLIN mapa a los puertos de Flowcode.

	7	6	5	4	3	2	1	0
PORTA	**13** (36)	12 (37)	5	4 (32)	26 (41)	2 (34)	33 (5)	32 (4)
PORTB	25 (40)	0 (33)	22	27 (39)	19	18	15 (35)	14 (38)
PORTC	39 (3)	36 (0)	35 (7)	34 (6)	17	16	NC	23

Sólo entrada

() Entrada analógica

Analógico disponible sólo cuando no se utiliza WiFi/Bluetooth

Todos los pines sólo toleran 3,3 voltios.
Cuando se establece por primera vez el poder mediante la conexión USB a la ESP32 LOLIN, para obtener el código en ejecución, el swich RST puede requerir brevemente presionado.

Anexo 3:

BTEC Nacional Nivel 3 Mapa de la unidad 6

A Investigar el hardware típico de un sistema microcontrolador			¿Cubierto?
A1 Hardware de control			
Capacidades de E/S: número, tipo (analógicas/digitales), puertos			✓
especificaciones de hardware: anchura del bus, velocidad del procesador			✓
memoria - RAM, ROM			✓
características del navegador - interruptores integrados, pila, PWM			□
periféricos necesarios			✓
coste y accesibilidad			□
facilidad de uso			□
software y lenguaje de programación			✓
tensiones de funcionamiento y requisitos de potencia			✓
A2 Dispositivos de entrada			
Entrada del usuario:			
	digital - interruptores y botones		✓
	analógico - potenciómetro de control		✓
Temperatura			
	Termistor		✓
	sensores de temperatura		✓
	sensor ambiental - temperatura y humedad		□
Luz			
	la resistencia dependiente de la luz (LDR)		✘
	IR - fototransistor, fotodiodo o receptor IR		✓
Movimiento/orientación			
	interruptor de inclinación		✘
Pres encia			
	mi cro-s bruja		✘
	Ultrasonidos		✘
Requisitos de la interfaz de entrada			
	s ignal conditioning		✓
	conversión analógico-digital (ADC)		✓
	tarjetas sensoriales modulares		✓
	PWM		✓
	comunicaciones en serie		□
	Ci rcui t o Inter-Integrado (I2C)		□
A3 Dispositivos de salida			
Optoe lectroni c			
	Diodo emisor de luz (LED) - indicador e IR		□
	P a n t a l l a d e 7 segmentos		✓
	pantalla de cristal líquido (LCD)		✓
Electromecánica			
	Relé		
	motor de corriente continua		
	Servo		
Audi o			
	zumbador o s i ren		
	altavoz o transductor piezoeléctrico		
Requisitos de interfaz de salida			
	requisitos de potencia y controladores		✓
		etapa de salida de transistor	
		Relé	
	PWM		✓
	comunicaciones en serie		
	Interfaz de dispositivos I2C		
A4 Selección de dispositivos de hardware y diseño del sistema			
A5 Montaje y funcionamiento de un sistema microcontrolador			

A Investigar el hardware típico de un sistema microcontrolador			¿Cubierto?
A1 Hardware de control			
Capacidades de E/S: número, tipo (analógicas/digitales), puertos			✓
especificaciones de hardware: anchura del bus, velocidad del procesador			✓
memoria - RAM, ROM			✓
funciones de hardware: interrupciones, pila, PWM			□
periféricos necesarios			✓
coste y accesibilidad			□
facilidad de uso			□
software y lenguaje de programación			✓
tensiones de funcionamiento y requisitos de potencia			✓
A2 Dispositivos de entrada			
Entrada del usuario:			
	digital - interruptores y botones		✓
	analógico - potenciómetro de control		✓
Temperatura			
	Termistor		✓
	sensores de temperatura		✓
	sensor ambiental - temperatura y humedad		□
Luz			
	l aresistencia dependiente delaluz(LDR)		✘
	IR - fototransistor, fotodiodo o receptor IR		✓
Movimiento/orientación			
	ti l t interruptor		✘
Presencia			
	microinterruptor		✘
	Ultrasonidos		✘
Requisitos de la interfaz de entrada			
	s ignal conditioning		✓
	conversión analógico-digital (ADC)		✓
	tarjetas sensoriales modulares		✓
	PWM		✓
	comunicaciones en serie		□
	Circuito integrado (I2C)		□
A3 Dispositivos de salida			
Optoelectrónica			
	Diodo emisor de luz (LED) - indicador e IR		□
	Pantalla de 7 segmentos		✓
	pantalla de cristal líquido (LCD)		✓
Electromecánica			
	Relé		
	motor de corriente continua		
	Servo		
Audio			
	zumbador o s i ren		
	altavoz o transductor piezoeléctrico		
Requisitos de interfaz de salida			
	requisitos de potencia y controladores		✓
		etapa de salida de transistor	
		Relé	
	PWM		✓
	comunicaciones en serie		
	Interfaz de dispositivos I2C		
A4 Selección de dispositivos de hardware y diseño del sistema			
A5 Montaje y funcionamiento de un sistema microcontrolador			

C Ciclo de desarrollo del sistema			**Proyecto**
C1 Procesos de desarrollo			**Proyecto**
Etapas del proceso de desarrollo.			
Documentación C2			**Proyecto**
Un expediente de pruebas			
durante todo el proceso de desarrollo.			

Control de versiones

29 11	21	Convertido en Publisher y actualizado. Versión 3.
19 09	23	Agregados algunos ejercicios en la sección Ejercicios adicionales usando sims Flowcode. Página 87, 88